Annex A (normative)
Transportable file structure

A.1 Transportable file structure (TFS)

The Transportable File Structure (TFS) Data Extension Element allows for configuration, data request, commands, and PIKS object data to be stored in hierarchy order with Metadata associated for each level. The TFS shall either contain data for the objects or provide a computer system an unambiguous reference to the data. PIKS objects can be stored in a TFS to perform the required image processing steps to a given image. While the TFS has the capability to express complicated structure relating several hierarchy objects, the TFS can relate a BIIF image with several PIKS objects in a simple expression. The TFS can be parsed using a state machine. Major sections of the TFS are delimited with TFS Delimiter commands. The initial state of the TFS is the TFS state. The TFS Descriptor commands describing the overall structure of the TFS, the TFS version Metadata, security, subscription, configuration, and an index into the TFS transports are found in this state. The next state of the TFS is the Transport State. The Transport Descriptor commands describe the security, Metadata, and index into the transport body components. Within the transport body the TFS enters one or more Profile states. The Profile Descriptor commands describe the security, Metadata, and index for each profile state. The profile body contains objects, actions, or other nested profiles. Profile objects contain data formatted according to object type. The object data may contain an unambiguous reference to data stored elsewhere, such as in BIIF file, or contain the actual object data. More than one profile can be contained within a transport body and the TFS can contain more than one transport. Furthermore, the profiles can be nested to create hierarchical structures. For the simple case of one BIIF image and several PIKS objects, these objects should be inside a single profile for processing. Each TFS, transport, and profile state contains security, Metadata, and an associated index into the next hierarchical level. The TFS escape state provides a mechanism for the insertion of application defined data. The TFS escape state terminates upon completion of the TFS escape command. All other states have a begin state and an end state.

A.1.1 TFS Commands

The following describes TFS commands grouped by command class. The delimiter commands change the TFS state. The descriptor commands provide data within each state.

1. TFS Delimiter Commands
   - BEGIN TFS
   - BEGIN TRANSPORT
   - BEGIN TRANSPORT BODY
   - BEGIN PROFILE
   - BEGIN PROFILE BODY
   - END PROFILE
   - END TRANSPORT
   - END TFS

2. TFS Descriptor Commands
   - TFS VERSION
   - TFS SECURITY
   - TFS METADATA
   - TFS INDEX
   - TFS SUBSCRIPTION
   - TFS CONFIGURATION
   - TFS CONFIG DATA

3. Transport Descriptor Commands
   - TRANSPORT SECURITY
   - TRANSPORT METADATA
   - TRANSPORT INDEX

4. Profile Descriptor Commands
   - PROFILE SECURITY
   - PROFILE METADATA
   - PROFILE INDEX

5. Profile Commands
   - PROFILE ACTION

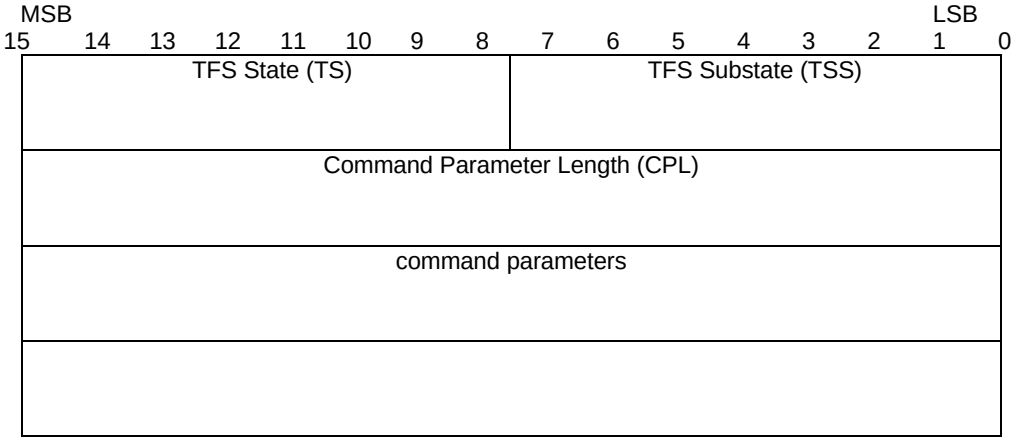
PROFILE OBJECT

6. TFS User Commands
   TFS ESCAPE

A.1.2 TFS encoding

Each TFS command is identified by the combination of the TFS State (TS) and TFS Substate (TSS) fields which comprise the first two bytes of the command. The identification is followed by two or four bytes called the Command Parameter Length (CPL) specifying the length of the balance of the command. The balance of the command is a series of parameters whose content depends upon the identity of the command. The TFS commands are encoded in binary form. The 8 most significant bits (MSB) of the first word identify the TFS state in which the command belongs. The 8 least significant bits (LSB) specify the TFS substate of the command. The next 2 bytes are interpreted as a signed positive integer containing the length and the CPL of the command parameters. Parameters are padded with a trailing ASCII null byte (0) to ensure that all subsequent commands begin on a 16-bit word boundary. The trailing null byte of the last parameter is not included in the parameter list length. If the CPL is greater than 32,767, then the CPL shall be two 16-bit words long with the MSB of the high order byte (word 1) set to 1. The value of the CPL is that of the remaining 31 bits in the two words and can contain values up to 999,999,999 bytes for the full CPL. Note, the TFS size must comply with the overall size of the DES data field in which it is included. Field numbers in tables contained in this annex are in hexadecimal notation.

Table A.1 -- Encoding of a TFS command



A.1.3 TFS command flow

The following flow diagram displays all the TFS commands contained in a Metafile. The commands are executed in sequential order. The TFS Descriptor Commands occur after the BEGIN TFS command and before the BEGIN TRANSPORT command. The Transport Body Descriptor Commands occur after the BEGIN TRANSPORT command and before the BEGIN TRANSPORT BODY command. The Profile Descriptor Commands occur after the BEGIN PROFILE command and before the BEGIN PROFILE BODY command. The Profile Commands occur after the BEGIN PROFILE BODY command and before the END PROFILE command. BEGIN PROFILE and associated commands can be nested. The TFS ESCAPE command can occur at any position in the TFS where any other command can occur except the BEGIN TFS and END TFS commands. The commands enclosed within matching brackets are optional within a given metafile.

BEGIN TFS
  TFS VERSION
  TFS SECURITY
  [TFS INDEX]
  [TFS SUBSCRIPTION]
  [TFS CONFIGURATION]
  [TFS CONFIG DATA]
  [TFS METADATA]
  [BEGIN TRANSPORT
      TRANSPORT SECURITY
      [TRANSPORT METADATA]

```
        [TRANSPORT INDEX]
        [BEGIN TRANSPORT BODY
            [BEGIN PROFILE
                PROFILE SECURITY
                [PROFILE METADATA]
                [PROFILE INDEX]
                [BEGIN PROFILE BODY
                        [PROFILE ACTION
                        [PROFILE OBJECT]
                        [BEGIN PROFILE
                            ...
                        END PROFILE]
            END PROFILE]]
    END TRANSPORT]
END TFS
```

A.2 TFS command formats

The following subclauses list the required TFS commands along with the encoding.

A.2.1 TFS delimiter commands

The TFS Delimiter commands define boundaries for significant structures within the TFS. The following table defines the TS, TSS, and the CPL for the TFS Delimiter commands. The value x for CPL indicates that a non- zero value shall be used if there is data supplied with the command.

Table A.2 -- TFS delimiter command table

TFS COMMAND NAME	TS	TSS	CPL
BEGIN TFS	10	1	x
BEGIN TRANSPORT	10	2	x
BEGIN TRANSPORT BODY	10	3	0
BEGIN PROFILE	10	4	x
BEGIN PROFILE BODY	10	5	0
END PROFILE	10	6	0
END TRANSPORT	10	7	0
END TFS	10	8	0

A.2.1.1 Begin TFS command

The BEGIN TFS command shall have non- zero values for CPL if it has parameter data. In this case, the parameter is a name associated with the TFS file. This name is represented using the character string C1, C2, ... Cn with length n. The following tables describes the even and odd format with x = n+1;0. No data will be in the length or character fields of the command format as depicted in Tables A.3 and A.4 when n and CPL equal 0.

Table A.3 -- TFS command even form

MSB 15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 LSB 0
TS	TSS
CPL = n+1;0	
length = n (even)	C1
C2	. . .
Cn	0

Table A.4 -- TFS command odd form

MSB 15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 LSB 0
TS	TSS
CPL = n+1;0	
length = n (odd)	C1
C2	. . .
C(n-1)	Cn

A.2.1.2 Begin transport command

The BEGIN TRANSPORT command shall have non- zero values for CPL if it has parameter data. In this case, the parameter is a name associated with the receiver of the transport data. This name is represented using the character string C1, C2, ... Cn with length n and conforms to the format depicted in Tables A.3 and A.4 with associated TS and TSS and non-zero CPL.

A.2.1.3 Begin transport body command

The BEGIN TRANSPORT BODY command conforms to the format depicted in Tables A.3 and A.4 with associated TS and TSS and zero CPL. See Tables A.3 and A.4.

A.2.1.4 Begin profile command

The BEGIN PROFILE command shall have non- zero values for CPL if it has parameter data. In this case, the parameter is a name associated with the profile. This name is represented using the character string C1, C2, ... Cn with length n and conforms to the same format as the Begin TFS command with associated TS and TSS and non zero CPL. See Tables A.3 and A.4.

A.2.1.5 Begin profile body command

The BEGIN PROFILE BODY command conforms to the same format as the Begin TFS command with associated TS and TSS and zero CPL.

A.2.1.6 End profile command

The END PROFILE command conforms to the same format as the Begin TFS command with associated TS and TSS and zero CPL.

A.2.1.7 End transport command

The END TRANSPORT command conforms to the same format as the Begin TFS command with associated TS and TSS and zero CPL.

A.2.1.8 End TFS command

The END TFS command conforms to the same format as the Begin TFS command with associated TS and TSS and zero CPL.

A.2.2 TFS descriptor commands

The TFS Descriptor commands describe the functional content, security, and characteristics of the TFS. The following table defines the TS, TSS, and the CPL for the TFS Descriptor commands. The value x for CPL indicates that a non-zero value shall be used if there is data supplied with the command. The value 2 for CPL indicates that the CPL value is 2.

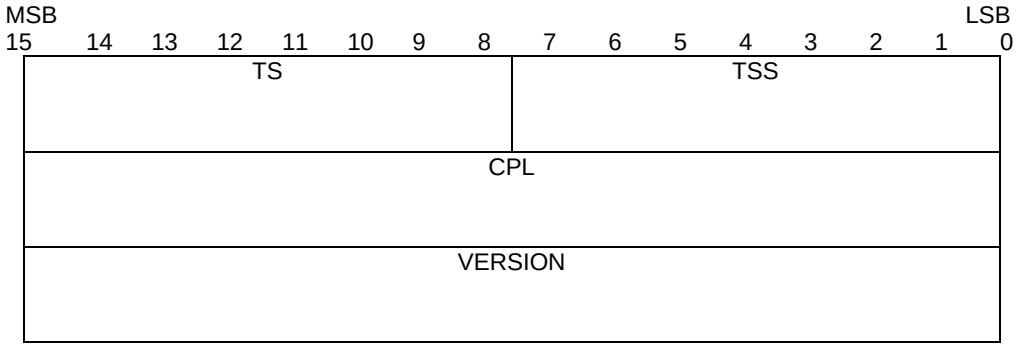
Table A.5 -- TFS descriptor command table

TFS COMMAND NAME	TS	TSS	CPL
TFS VERSION	20	1	2
TFS SECURITY	20	2	x
TFS SUBSCRIPTION	20	3	x
TFS CONFIGURATION	20	4	2
TFS CONFIG DATA	20	5	x
TFS METADATA	20	6	x
TFS INDEX	20	7	x

A.2.2.1 TFS version command

The following is the TFS VERSION command format. The VERSION shall be an integer.

Table A.6 -- TFS version



A.2.2.2 TFS security command

The TFS SECURITY command conforms to the same format as the TFS Delimiter commands with associated TS and TSS and non-zero CPL.

A.2.2.3 TFS subscription command

The following is the TFS SUBSCRIPTION command format. The SUBSCRIPTION value is 1 to subscribe to data objects, 2 for a data pull, and 3 to cancel subscription. Each subscription service consists of an unsigned 8-bit service name length followed by a character name. Each character name shall be in the form "name = identifier".

Table A.7 -- TFS subscription command

MSB 15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 LSB 0
TS	TSS
CPL	
SUBSCRIPTION	
length = a	C1
C2	. . .
Ca	. . .
length = b	C1
C2	. . .
Cb	. . .

The following table describes the contents for the TFS Subscription command. Each command is of the form "name = identifier". The URL identifier determines how to send the TFS to the subscriber. The OBJECT-NAME identifier determines how the TFS objects are to be stored.

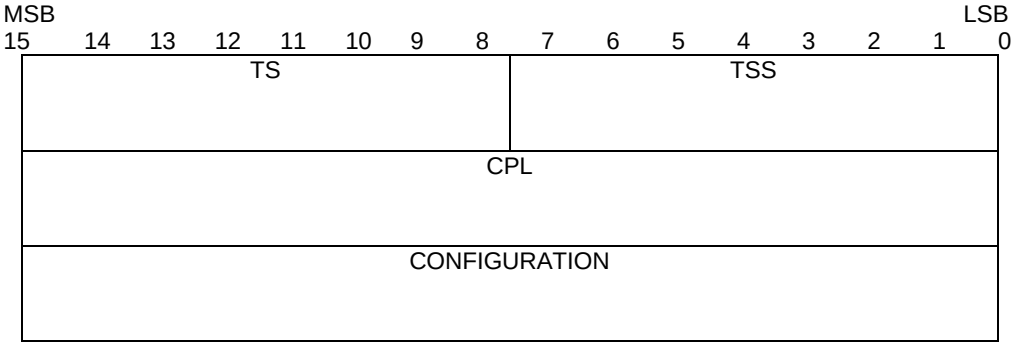
Table A.8 -- TFS subscription contents table

NAME	IDENTIFIER
USERNAME	Login Name
PASSWORD	Login Password
HOSTNAME	Computer Host Name or IP Address
URL	URL identifier - What transfer to use to deliver the TFS (E-MAIL, FTP, HTTP, ASYNCHRONOUS, etc.)
DIRECTORY	Directory where TFS is to be deposited
FILENAME	File name to use when depositing the TFS in above directory
OBJECT_NAME	How object is to be stored 0=METADATA ONLY - No object data 1=TFS - object data stored in TFS 2 =URL - TFS string point to object data

A.2.2.4 TFS configuration command

The following is the TFS CONFIGURATION command format. The CONFIGURATION value is 0 to provide configuration and 1 to request configuration. Note, if the CONFIGURATION value is 0 then the TFS CONFIG DATA command is required.

Table A.9 -- TFS configuration command



A.2.2.5 TFS configuration data command

The following is the TFS CONFIG DATA command format. Each configuration service consists of an unsigned 8-bit service name length followed by the configuration service name. This command is supplied by a subscriber when the TFS Configuration Command CONFIGURATION value 0 and supplied to a subscriber upon request. Each configuration service name shall be in the form "OBJECT_NAME=service" where OBJECT_NAME is a valid object name to be used in the TFS and service is the required program to execute for that object.

Table A.10 -- TFS config data command

MSB 15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 LSB 0
TS	TSS
CPL	
length = a	C1
C2	. . .
Ca	. . .
length = b	C1
C2	. . .
Cb	. . .

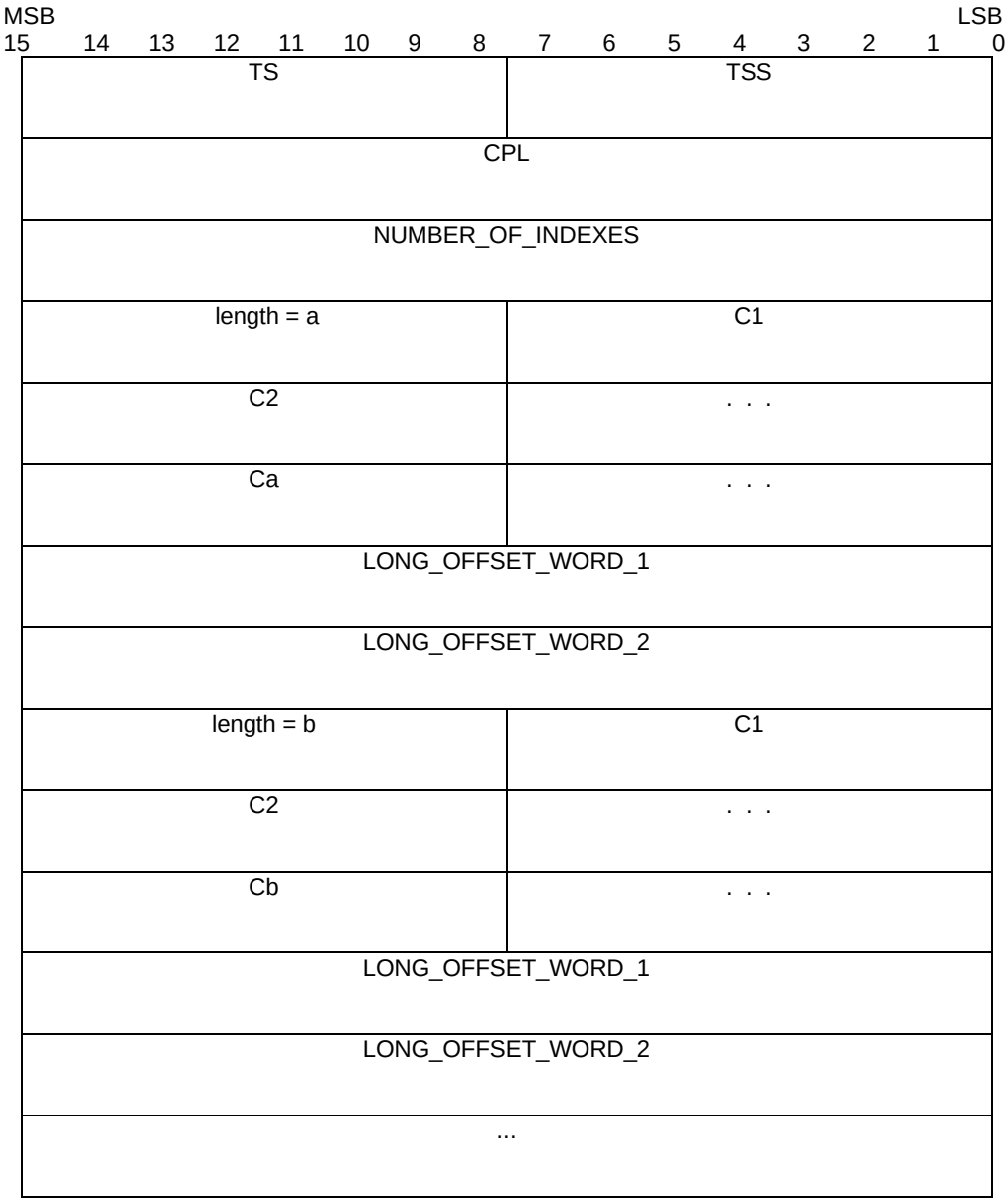
A.2.2.6 TFS metadata command

The TFS METADATA command conforms to the same format as the TFS Delimiter commands with associated TS and TSS and non-zero CPL. See Tables A.3 and A.4.

A.2.2.7 TFS index command

The TFS INDEX command provides a quick entry into the Transports of the TFS. The NUMBER_OF_INDEXES field is an integer field that specifies the number of indexes, one for each BEGIN TRANSPORT command. Each index consists of an unsigned 8-bit index name length followed by the character index name followed by the index. The name must match the name contained in the BEGIN TRANSPORT parameter to which the index points. The index offset is expressed in bytes and is calculated relative to the BEGIN TFS command (beginning of the TFS) whose first byte has a value of 1 to the BEGIN TRANSPORT command. There must be an entry for each BEGIN TRANSPORT command in the TFS and the index must be to and inclusive of the first byte of those commands.

Table A.11 -- TFS index command



A.2.3 Transport descriptor commands

The Transport Body Descriptor Commands describe the functional content, security, and characteristic of the transport body. The following table defines the TS, TSS, and the CPL for the Transport Descriptor commands. The value x for CPL indicates that a non-zero value shall be used if there is data supplied with the command.

Table A.12 -- Transport descriptor command table

TFS COMMAND NAME	TS	TSS	CPL
TRANSPORT SECURITY	30	1	x
TRANSPORT METADATA	30	2	x
TRANSPORT INDEX	30	3	x

A.2.3.1 Transport security command

The TRANSPORT SECURITY command conforms to the same format as the TFS Delimiter commands with associated TS and TSS and non-zero CPL.

A.2.3.2 Transport metadata command

The TRANSPORT METADATA command conforms to the same format as the TFS Delimiter commands with associated TS and TSS and non-zero CPL. See Tables A.3 and A.4.

A.2.3.3 Transport index command

The TRANSPORT INDEX command provides a quick entry into the results contained in the Transport Body. The TRANSPORT INDEX command conforms to the same format as the TFS INDEX command with associated TS, TSS, and non-zero CPL. The NUMBER_OF_INDEXES field is an integer field that specifies the number of indexes, one for each BEGIN_PROFILE command. Each index consists of an unsigned 8-bit index name length followed by the character index name followed by the index. The name must match the name contained in the BEGIN PROFILE parameter to which the index points. The index offset is expressed in bytes and is calculated relative to the BEGIN TRANSPORT command whose first byte has a value of 1. There must be an entry for each BEGIN PROFILE command in the Transport Body, and the index must be to and inclusive of the first byte of those command. See Table A.11.

A.2.4 Profile descriptor commands

The Profile Descriptor commands describe the functional content, security, and characteristic of the profile. The following table defines the TS, TSS, and the CPL for the Profile Descriptor commands. The value x for CPL indicates that a non-zero value shall be used if there is data supplied with the command

Table A.13 -- Profile descriptor command table

TFS COMMAND NAME	TS	TSS	CPL
PROFILE SECURITY	40	1	x
PROFILE METADATA	40	2	x
PROFILE INDEX	40	3	x

A.2.4.1 Profile security command

The PROFILE SECURITY command conforms to the same format as the TFS Delimiter commands with associated TS and TSS and non-zero CPL. See Tables A.3 and A.4.

A.2.4.2 Profile metadata command

The PROFILE METADATA command conforms to the same format as the TFS Delimiter commands with associated TS and TSS and non-zero CPL. See Tables A.3 and A.4.

A.2.4.3 Profile index command

The PROFILE INDEX command provides a quick entry into the Actions, Objects, and embedded Profiles contained in the Profile Body. The PROFILE INDEX command conforms to the same format as the TFS INDEX command with associated TS, TSS, and non-zero CPL. The NUMBER_OF_INDEXES field is an integer field that specifies the number of indexes, one for each PROFILE ACTION, PROFILE OBJECT, and BEGIN PROFILE command. Each index consists of an unsigned 8-bit index name length followed by the character index name followed by the index. The name must match the name contained in the PROFILE ACTION, PROFILE OBJECT, or BEGIN PROFILE parameters to which the index points. The index offset is expressed in bytes and is calculated relative to the BEGIN PROFILE command whose first byte has a value of 1. There must be an entry for each PROFILE COMMAND, PROFILE OBJECT, and BEGIN PROFILE commands in the Profile Body, and the index must be to and inclusive of the first byte of those commands. See Table A.11.

A.2.5 Profile commands

The Profile commands describe the individual objects contained in each profile. The following table defines the TS, TSS, and the CPL for the Profile commands. The value x for CPL indicates that a non-zero value shall be used if there is data supplied with the command.

Table A.14 -- Profile command table

TFS COMMAND NAME	TS	TSS	CPL
PROFILE ACTION	50	1	x
PROFILE OBJECT	50	2	x

A.2.5.1 Profile action command

The PROFILE ACTION command conforms to the same format as the TFS Delimiter commands with associated TS and TSS and non-zero CPL. See Tables A.3 and A.4. The command shall be in the form "OBJECT_NAME=value" where command_name and value are defined by the profile.

A.2.5.2 Profile object command

The PROFILE OBJECT command provides a mechanism to reference all-source objects inside a profile for processing. Each object has a type followed by a name and then the specified data or a pointer to the data. The following is the PROFILE OBJECT command format. The OBJECT_TYPE field is an integer that specifies the type of data. If OBJECT_TYPE=1, then the object storage is given by the TFS Configuration Data command by the configuration service "OBJECT_NAME=action_storage". If OBJECT_TYPE=2, then the object data is contained as a BIIF component, and that component number is given in the OBJECT_DATA field. The next field is the length of the OBJECT_NAME followed by the OBJECT_NAME in characters (C1, C2, ..., Ca). The OBJECT_STORAGE field is an integer that determines how the data is stored. If OBJECT_STORAGE=0 then the object data is contained in the OBJECT_DATA field or is a BIIF component. If OBJECT_STORAGE does not equal 0, then the storage for the object data is given in the OBJECT_DATA field and was determined by the TFS Subscription command.

Table A.15 -- Profile object command

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
TS								TSS							
CPL															
OBJECT_TYPE															
length = a								C1							
C2								. . .							
Ca								. . .							
OBJECT_STORAGE															
OBJECT_DATA															
...															

The following table describes the contents of the OBJECT_DATA field for the OBJECT_TYPE=2 (BIIF segment). The first entry is the name of the BIIF segment identifier (IM, SY, TE, DE, or RE) concatenated with the segment name (IID, SID, TEXTID, DESTAG, or RESTAG). The second entry is the integer value for the BIIF_SEGMENT. The BIIF_SEGMENT points to the associated Display Level (1-999) in the BIIF file for images and symbols. When the BIIF_SEGMENT is a TEXT, DES, or a RES segment, the BIIF_SEGMENT is the sequence number of occurrence for the given standard segment type.

Table A.16 -- Object data for BIIF profile object

MSB 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	LSB 0
length = a								C1							
C2								. . .							
Ca								. . .							
BIIF_SEGMENT															

The following table describes the contents for the OBJECT_STORAGE and associated OBJECT_DATA fields for the PROFILE OBJECT command.

Table A.17 -- Object storage and object data format

OBJECT_STORAGE	OBJECT_DATA (FORMAT)
0=METADATA ONLY	No object data
1=TFS	Object data stored in TFS or BIIF
2=URL	TFS string pointing to object data

A.2.5.2.1 Profile object command parameters for PIKS objects

The TFS provides the capability to include PIKS Objects for image processing. PIKS Objects are associated with an image using the BIIF_SEGMENT as defined in clause A.2.5.2, Profile object command. Both the Profile object command containing the BIIF_SEGMENT and the Profile containing the PIKS Objects are embedded inside a single Profile to define the relationship. The following table defines the minimum size for the PIKS data types used for PIKS objects.

Table A.18 – PIKS Objects minimum size

DATATYPE	MINIMUM SIZE IN BITS	C DATATYPE (INFORMATIVE)
BP	8	UNSIGNED CHAR
NP	64	UNSIGNED LONG INT
SP	32	INT
RP	32	FLOAT
CP	64	FLOAT PAIR
CS	8	CHAR
BD	1	UNSIGNED CHAR
ND	8	CHAR
SD	16	SHORT
RD	32	FLOAT
CD	64	FLOAT PAIR

The following table defines the OBJECT_DATA field contents for the OBJECT_TYPE of PIKS. The name of the PIKS object shall be presented as the first parameter as a character string. The second parameter shall be a character string specifying the object identifier. The following parameters and lengths are defined in the table below.

Table A.19 -- PIKS objects data structures

PIKS OBJECT	Parameters	Length
OBJECT_CLASS “PIKS_IMAGE”	CS	10
PIKS_IMAGE_ID	CS	Max 32
X_SIZE	NP	1
Y_SIZE	NP	1
Z_SIZE	NP	1
T_SIZE	NP	1
B_SIZE	NP	1
X_BANDTYPE	NP	1
Y_BANDTYPE	NP	1
Z_BANDTYPE	NP	1
T_BANDTYPE	NP	1
B_BANDTYPE	NP	1
X_WHITE_POINT	RP	1
Y_WHITE_POINT	RP	1
Z_WHITE_POINT	RP	1
STRUCTURE	SP	1
COLOUR_SPACE	SP	1
DATA Note - For this object the image data is located in the image segment having this display level Identified in this BIIF_ATTACHMENT_LEVEL associated with this TFS	BD,ND,SD,RD,CD	[X_SIZE]*[SIZEOF(X_BANDTYPE)]* [Y_SIZE]*[SIZEOF(Y_BANDDYPE)]* [Z_SIZE]*[SIZEOF(Z_BANDTYPE)]* [T_SIZE]*[SIZEOF(T_BANDTYPE)]* [B_SIZE]*[SIZEOF(B_BANDTYPE)]

Table A.19 -- PIKS objects data structures *(continued)*

PIKS OBJECT	Parameters	Length
OBJECT_CLASS “PIKS_HISTOGRAM”	CS	14
HISTOGRAM_ID	CS	Max 32
SIZE	NP	1
LOWER_BOUND	RP	1
UPPER_BOUND	RP	1
DATA	NP	[SIZE]*[SIZEOF(NP)]

Table A.19 -- PIKS objects data structures *(continued)*

PIKS OBJECT	Parameters	Length
OBJECT_CLASS “PIKS_LOOKUP_TABLE”	CS	Max 32
LUT_ID	CS	Max 32
ENTRIES	NP	1
BANDS	NP	1
INPUT_DATATYPE	SP	1
OUTPUT_DATATYPE	SP	1
DATA	BD,ND,SD,RD,CD	[ENTRIES]*[BANDS]* [SIZEOF(OUTPUT_DATATYPE)]

Table A.19 -- PIKS objects data structures *(continued)*

PIKS OBJECT	Parameters	Length
OBJECT_CLASS “PIKS_MATRIX”	CS	Max 32
MATRIX_ID	CS	Max 32
COLUMNS	NP	1
ROWS	NP	1
DATATYPE	SP	1
DATA	ND,SD,RD,CD	[COLUMNS]*[ROWS]* [SIZEOF(DATATYPE)]

Table A.19 -- PIKS objects data structures *(continued)*

PIKS OBJECT	Parameters	Length
OBJECT_CLASS “PIKS_NEIGHBOURHOOD”	CS	18
ARRAY_ID	CS	Max 32
X_SIZE	NP	1
Y_SIZE	NP	1
Z_SIZE	NP	1
T_SIZE	NP	1
B_SIZE	NP	1
X_KEY_PIXEL	SP	1
Y_KEY_PIXEL	SP	1
Z_KEY_PIXEL	SP	1
T_KEY_PIXEL	SP	1
B_KEY_PIXEL	SP	1
SCALE_FACTOR	SP	1
LABEL	SP	1
DATA_TYPE	SP	1
DATA	BD,ND,SD,RD,CD	[X_SIZE]*[Y_SIZE]*[Z_SIZE]* [T_SIZE]*[B_SIZE]* [SIZEOF(DATATYPE)]

Table A.19 -- PIKS objects data structures *(continued)*

PIKS OBJECT	Parameters	Length
OBJECT_CLASS “PIKS_PIXEL_RECORD”	CS	17
RECORD_ID	CS	Max 32
B	NP	1
BAND_1_DATATYPE	SP	1
BAND_2_DATATYPE	SP	1
…	…	…
BAND_B_DATATYPE	SP	1
DATA	BD,ND,SD,RD,CD	[SIZEOF(BAND_1_DATATYPE)]* [SIZEOF(BAND_2_DATATYPE)]*…* [SIZEOF(BAND_B_DATATYPE)]

Table A.19 -- PIKS objects data structures *(continued)*

PIKS OBJECT	Parameters	Length
OBJECT_CLASS “PIKS_STATIC”	CS	Max 32
STATIC_ID	CS	Max 32
X_SIZE	NP	1
Y_SIZE	NP	1
Z_SIZE	NP	1
T_SIZE	NP	1
B_SIZE	NP	1
LABEL	SP	1
DATATYPE	SP	1
DATA	BD,ND,SD,RD,CD	[X_SIZE]*[Y_SIZE]*[Z_SIZE]* [T_SIZE]*[B_SIZE]* [SIZEOF(DATATYPE)]

Table A.19 -- PIKS objects data structures *(continued)*

PIKS OBJECT	Parameters	Length
OBJECT_CLASS “PIKS_TUPLE”	CS	10
TUPLE_ID	CS	Max 32
ENTRIES	NP	1
DATATYPE	SP	1
DATA	BD,ND,SD,RD,CD,CS	[ENTRIES]*[SIZEOF(DATATYPE)]

Table A.19 -- PIKS objects data structures *(continued)*

PIKS OBJECT	Parameters	Length
OBJECT_CLASS “PIKS_ROI_ARRAY”	CS	Max 32
ROI_ARRAY_ID	CS	Max 32
X_SIZE	NP	1
Y_SIZE	NP	1
Z_SIZE	NP	1
T_SIZE	NP	1
B_SIZE	NP	1
DATA	BP	[[X_SIZE]*[Y_SIZE]*[Z_SIZE]* [T_SIZE]*[B_SIZE]] / [8]

Table A.19 -- PIKS objects data structures *(continued)*

PIKS OBJECT	Parameters	Length
OBJECT_CLASS “PIKS_ROI_COORDINATE”	CS	19
ROI_COORDINATE_ID	CS	Max 32
X_SIZE	NP	1
Y_SIZE	NP	1
Z_SIZE	NP	1
T_SIZE	NP	1
B_SIZE	NP	1
COLLECTION_SIZE	NP	1
POLARITY	NP	1
DATA	NP	[SIZE]*[5]*[SIZEOF(NP)]

Table A.19 -- PIKS objects data structures *(continued)*

PIKS OBJECT	Parameters	Length
OBJECT_CLASS “PIKS_ROI_ELLIPTICAL”	CS	19
ROI_ELLIPTICAL_ID	CS	Max 32
X_SIZE	NP	1
Y_SIZE	NP	1
Z_SIZE	NP	1
T_SIZE	NP	1
B_SIZE	NP	1
X_ELLIPSE_CENTRE	NP	1
Y_ELLIPSE_CENTRE	NP	1
Z_ELLIPSE_CENTRE	NP	1
T_ELLIPSE_CENTRE	NP	1
B_ELLIPSE_CENTRE	NP	1
X_ELLIPSE_LENGTH	NP	1
Y_ELLIPSE_LENGTH	NP	1
Z_ELLIPSE_LENGTH	NP	1
T_ELLIPSE_LENGTH	NP	1
B_ELLIPSE_LENGTH	NP	1
X_INDEX_MANIPULATE	NP	1
Y_INDEX_MANIPULATE	NP	1
Z_INDEX_MANIPULATE	NP	1
T_INDEX_MANIPULATE	NP	1
B_INDEX_MANIPULATE	NP	1
DIMENSIONALITY	NP	1
POLARITY	NP	1

Table A.19 -- PIKS objects data structures *(continued)*

PIKS OBJECT	Parameters	Length
OBJECT_CLASS “PIKS_ROI_POLYGON”	CS	16
ROI_POLYGON_ID	CS	Max 32
X_SIZE	NP	1
Y_SIZE	NP	1
Z_SIZE	NP	1
T_SIZE	NP	1
B_SIZE	NP	1
ENTRIES	NP	1
X_INDEX_MANIPULATE	NP	1
Y_INDEX_MANIPULATE	NP	1
Z_INDEX_MANIPULATE	NP	1
T_INDEX_MANIPULATE	NP	1
B_INDEX_MANIPULATE	NP	1
POLARITY	NP	1
DATA	NP	[ENTRIES]*[2]*[SIZEOF(NP)]

Table A.19 -- PIKS objects data structures *(continued)*

PIKS OBJECT	Parameters	Length
OBJECT_CLASS “PIKS_ROI_RECTANGULAR”	CS	20
ROI_RECTANGULAR_ID	CS	Max 32
X_SIZE	NP	1
Y_SIZE	NP	1
Z_SIZE	NP	1
T_SIZE	NP	1
B_SIZE	NP	1
X_START	NP	1
Y_START	NP	1
Z_START	NP	1
T_START	NP	1
B_START	NP	1
X_END	NP	1
Y_END	NP	1
Z_END	NP	1
T_END	NP	1
B_END	NP	1
X_INDEX_MANIPULATE	NP	1
Y_INDEX_MANIPULATE	NP	1
Z_INDEX_MANIPULATE	NP	1
T_INDEX_MANIPULATE	NP	1
B_INDEX_MANIPULATE	NP	1
DIMENSIONALITY	NP	1
POLARITY	NP	1

Table A.19 -- PIKS objects data structures *(continued)*

PIKS OBJECT	Parameters	Length
OBJECT_CLASS “PIKS_VALUE_BOUNDS”	CS	17
BOUNDS_ID	CS	Max 32
SIZE	NP	1
LOWER_BOUND	SP	1
UPPER_BOUND	SP	1
DATATYPE	SP	1
DATA	NP,ND,SD,RD	[SIZE]*[[5]*[SIZEOF(NP)]+ [SIZEOF(DATATYPE)]]

Table A.19 -- PIKS objects data structures *(concluded)*

PIKS OBJECT	Parameters	Length
OBJECT_CLASS “PIKS_IMAGE”	CS	10
PIKS_IMAGE_ID	CS	Max 32
X_SIZE	NP	1
Y_SIZE	NP	1
Z_SIZE	NP	1
T_SIZE	NP	1
B_SIZE	NP	1
X_BANDTYPE	NP	1
Y_BANDTYPE	NP	1
Z_BANDTYPE	NP	1
T_BANDTYPE	NP	1
B_BANDTYPE	NP	1
X_WHITE_POINT	RP	1
Y_WHITE_POINT	RP	1
Z_WHITE_POINT	RP	1
STRUCTURE	SP	1
COLOUR_SPACE	SP	1
DATA Note - For this object the image data is located in the BIIF_ATTACHMENT_LEVEL associated with this TFS profile.	BD,ND,SD,RD,CD	[X_SIZE]*[SIZEOF(X_BANDTYPE)]* [Y_SIZE]*[SIZEOF(Y_BANDDYPE)]* [Z_SIZE]*[SIZEOF(Z_BANDTYPE)]* [T_SIZE]*[SIZEOF(T_BANDTYPE)]* [B_SIZE]*[SIZEOF(B_BANDTYPE)]

A.2.6 TFS escape command

The TFS ESCAPE command conforms to the generic TFS command format with TS = 60, TSS = 10, and non-zero CPL. This is an application-specific command that is to be defined by an application profile.

Annex B (normative)
Vector Quantization

B.1 Vector Quantized Data

Vector quantization (VQ) is a structuring algorithm chosen for use on multiband, color, and gray scale raster scanned maps and imagery because it provides predictable, rapid image reconstruction results. All information required for reconstruction of an BIIF VQ file is contained within the BIIF file itself. The concept of VQ is to represent monochrome or color image blocks with representative kernels from a code book. The indices of the representative kernels replace the image data in the quantized image. The code book and the color lookup table (LUT) are included in the file as overhead information.

B.2 Quantization Process

The VQ algorithm examines each v rows x h columns (v x h) pixel kernel in the input image and uses a clustering technique to develop a limited code book that contains the most representative kernels. The code book entries are v x h pixel kernels. These kernels are interpreted different ways, depending on the type of image they represent. In the case of Red, Green, Blue (RGB/LUT) quantized images, these pixels are actually indices into a color LUT. In other cases, they may represent indices to the gray scale pixel values or spectral band pixel values depending on formats of the gray scale (n-bit), color (RGB/LUT), color bands, R, G, B or multispectral bands. Figure B.1 shows the process for vector quantization. The procedure produces the code book and color LUT, if applicable, as part of the VQ header at the beginning of the image data field of the BIIF file, as illustrated in Figure B.2.

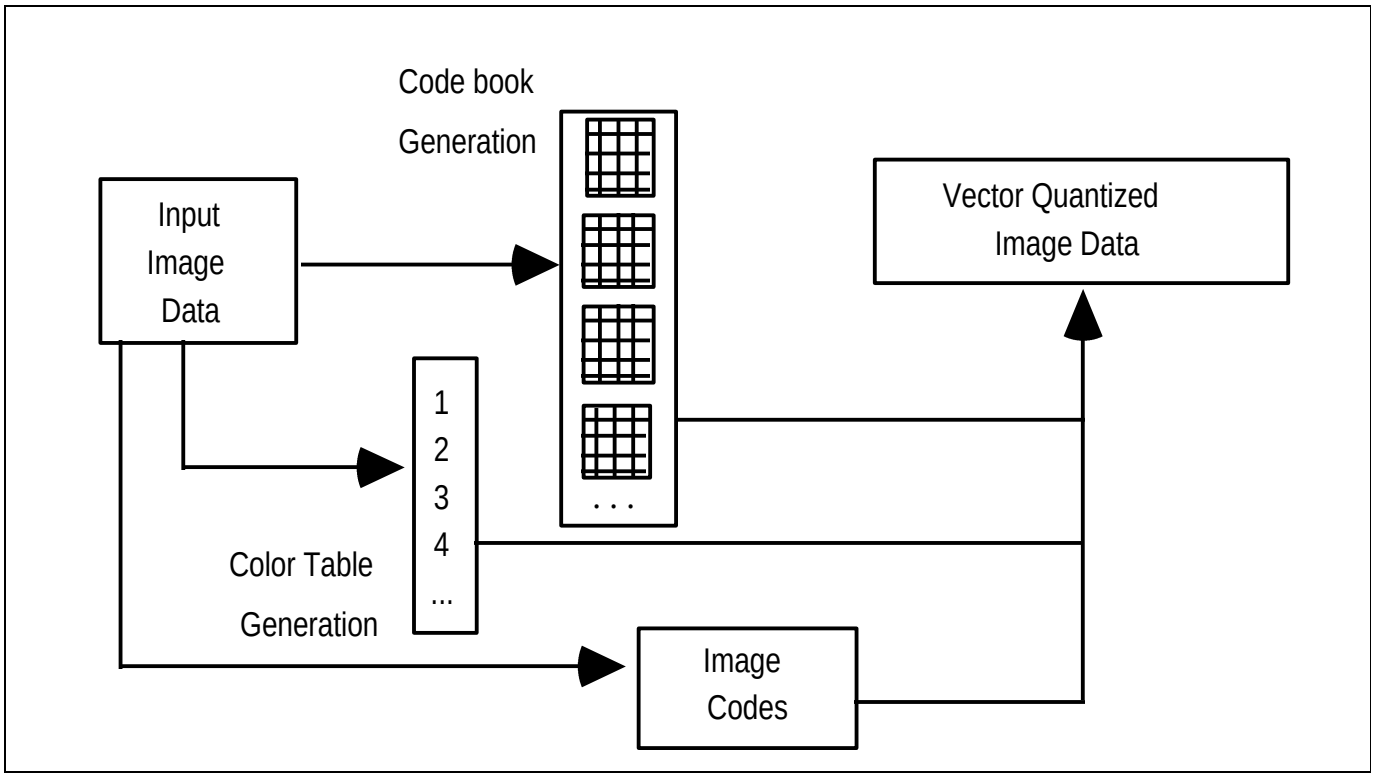


Figure B.1 -- Vector quantization process flow

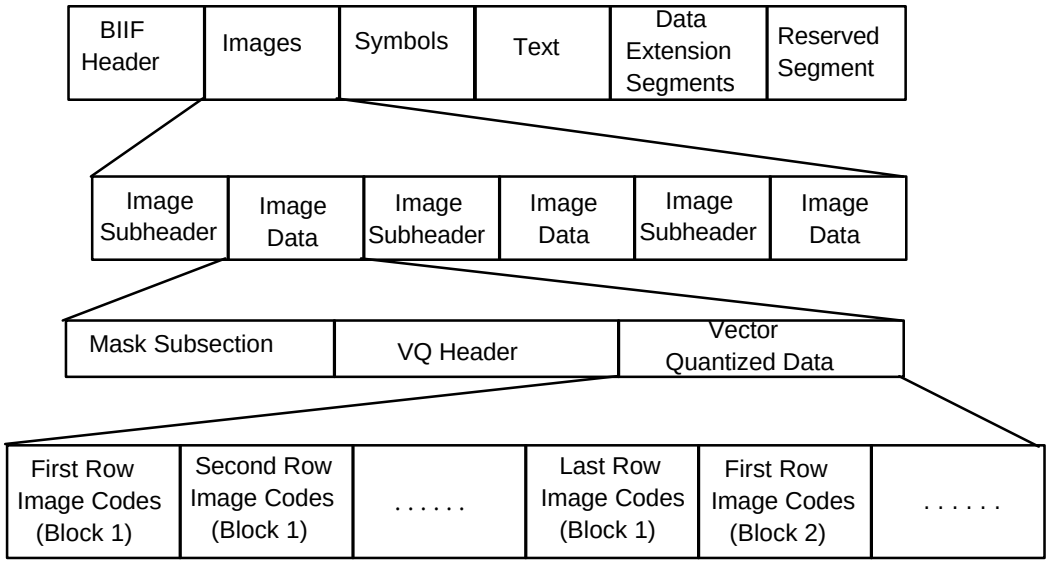


Figure B.2 -- BIIF file structure with VQ data

B.3 Reconstruction

The VQ data requires only a series of table lookups to reconstruct the image for display. The reconstruction process takes as input the quantized image data, which includes the image codes, code book(s), and color table (when applicable), and by a specified procedure, generates as its output digital reconstructed image data. This standard does not limit the implementation of VQ within BIIF in terms of the types and sizes of color lookup tables allowed. However, current implementation of VQ within BIIF uses a single RGB/LUT. Other organizations may be implemented in the future.

VQ reconstruction involves replacing image codes in the quantized image with pixel values for use in display or exploitation of the data. If the image has an associated LUT, the reconstruction is performed using the full process, as shown in Figure B3. The image reconstruction is complete at the first step if the quantized image does not have a color LUT. Color reconstruction would not be necessary in cases where the intended output pixel values are placed into the code book. This may occur in the encoding of gray scale imagery or in the encoding of multispectral imagery where each band is quantized separately.

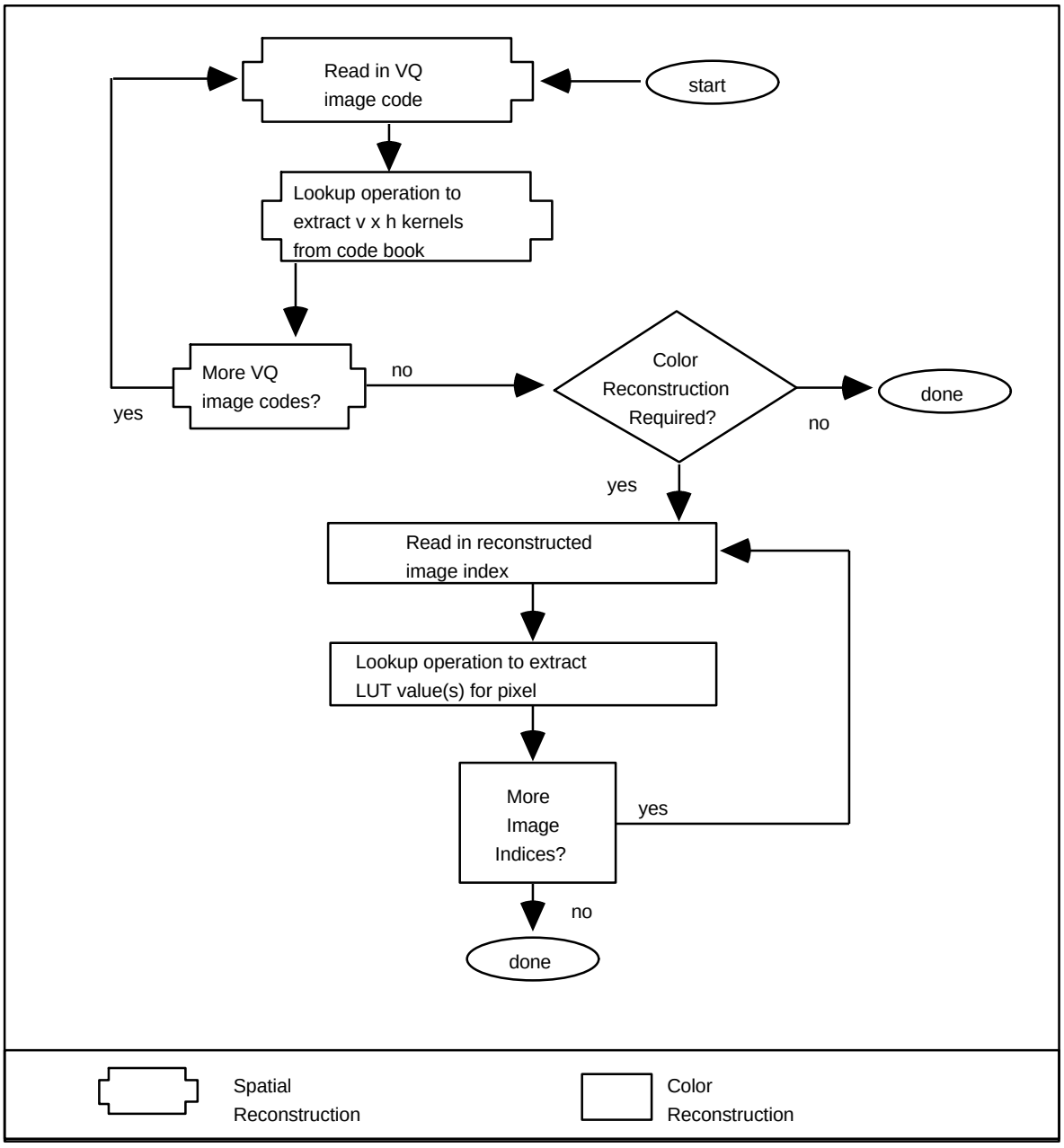


Figure B.3 -- VQ reconstruction procedure

B.3.1 Spatial reconstruction

When the BIIF IC field is set to C4 or M4, the image data field of the VQ formatted BIIF file shall contain a VQ header followed by the quantized image data. The VQ header shall contain information about the data including mask information (M4), and information defining the structure of the compressed image and code book. The code book that is used to reconstruct the image is also contained in the VQ header.

The code book within the image data consists of an array of image codes. Each image code is an index to the code book that has been constructed for the image. Each code book entry logically represents a group of v x h pixel indices. The BIIF structure

allows for the organization of the VQ code book to be optimized for the specific use of the VQ data. While some BIIF VQ products may require the VQ code book be arranged into v x h index kernels, other products may require that the individual rows for all v x h kernels be stored together such that the image can be reconstructed line-by-line, instead of kernel-by-kernel.

Each of the image codes, during VQ reconstruction, is converted to a kernel (or series of rows) of reconstructed pixel indices. The first image code appearing in the VQ image data field shall be used to spatially reconstruct the v x h indices in the upper left corner of the image. The reconstruction shall continue from left to right across the columns of the first row of image codes, then down each of the rows of image codes sequentially. The output is a spatially decoded image block. If the image has been color quantized, each value in the spatially reconstructed image represents an index into the color table. Figure B4 shows an example of the spatial reconstruction process. Various shades of gray are used to indicate higher or lower values in the code book. If the image is not color quantized, these values would be used to create a gray scale image where higher values in the code book typically correspond to brighter displayed pixels. For a color image, the values in the spatially reconstructed image correspond to indices in the LUT.

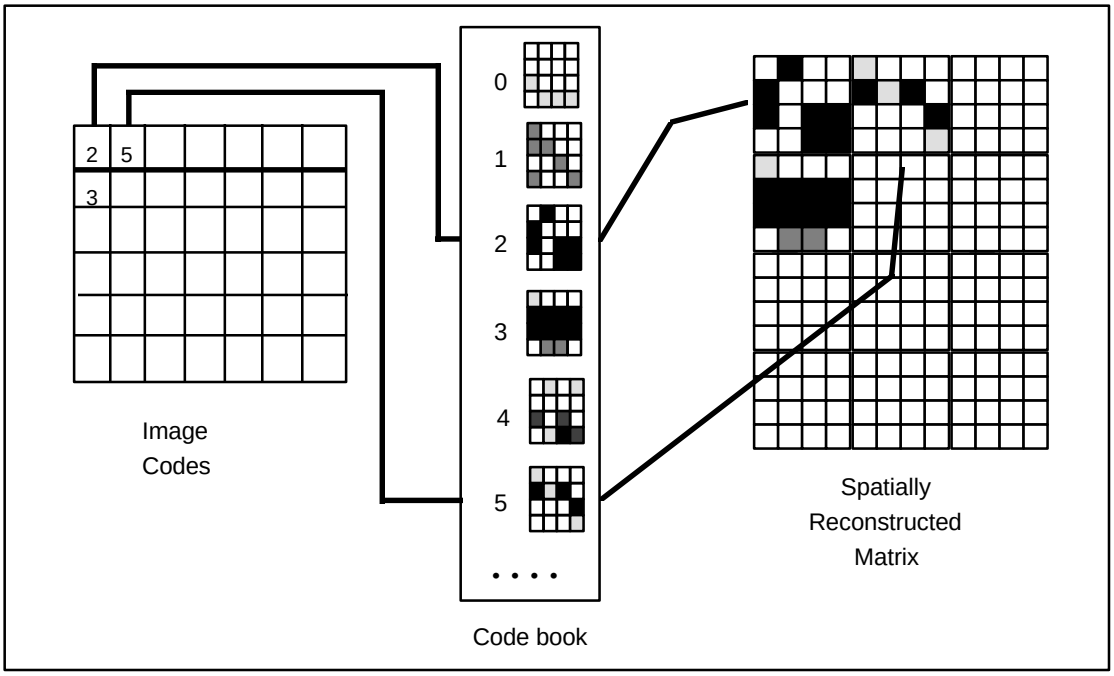


Figure B.4 -- Spatial reconstruction

B.3.2 Color reconstruction

Current implementation of VQ within BIIF has a limited scope and uses a single RGB/LUT. The output from the spatial data reconstruction process is an array consisting of values that represent either (1) monochromatic (gray scale) values for an image that is not color coded or (2) indices to the LUT in the BIIF image subheader if the image requires the use of a LUT. The final reconstruction step for color quantized images shall transform the indices into the corresponding pixel values by using the LUT values, illustrated in Figure B.5

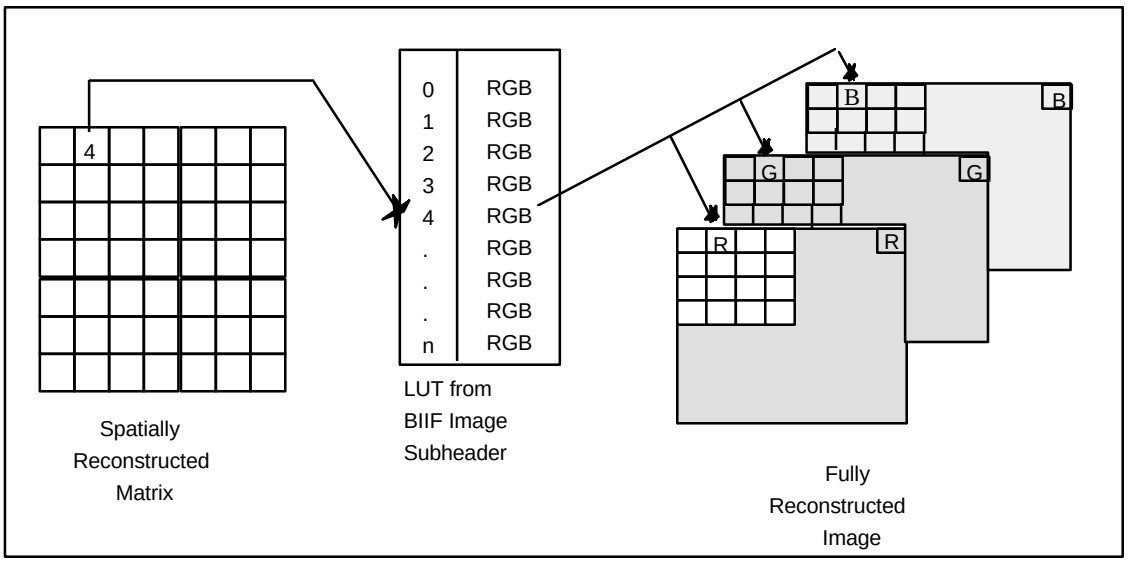


Figure B.5 -- Color reconstruction

B.3.3 Data elements

The BIIF VQ format allows for many quantization ratios recorded in the COMRAT field, and several organizations of the code books and color tables. The BIIF VQ file contains the information that the user needs in order to understand the organization of the data and to reconstruct the data for display. The following subclauses describe the fields in the BIIF VQ file that shall be used to determine the VQ organization of a particular file. Further information can also be taken from the vector quantization references provided in the body of this standard. Note: In order to work within the BIIF format structure, vector quantized data elements sometimes use compression/decompression terms in effort to maintain configuration within the BIIF.

B.3.3.1 Quantization ratio

Formulae for theoretical and actual vector quantization ratios are provided in the reference documents, and the results are entered in the BIIF Compression Ratio (COMRAT) field of the image subheader. This entry in a VQ file is generalized information and is not used in the reconstruction process. All BIIF VQ files shall contain a value in COMRAT given in the form n.nn representing the average number of bits-per-pixel for the image after vector quantization.

B.3.3.2 Masked vs unmasked

For vector quantized images, the Image Construct (IC) field of the image subheader shall contain the value C4 if the image is not masked or M4 if the image is masked. These codes are fully defined in the field definitions of the base BIIF standard.

B.3.3.3 Code book organization

The BIIF VQ image data subclause contains a decompression section where the VQ code book organization is defined. The data includes number of codes in the code book, the size of each v x h kernel, and how the data that make up the kernels are organized. The number of entries in the code book is represented in the <number of decode lookup records> field.

To determine how many pixels make up each kernel, the <number of image rows> field and <number of image codes per row> fields are employed, along with the number of pixels per block vertical (BIIF image subheader field NPPBV) and number of pixels per block horizontal (BIIF image subheader field NPPBH). The following equation is used to determine the size of the kernel in pixels:

$$v = \frac{\text{NPPBV}}{\text{<number of image rows>}} \qquad h = \frac{\text{NPPBH}}{\text{<number of image codes per row>}}$$

kernel size = v rows x h columns

The <number of decode lookup offset records> within the structure shall equal 1 if the data is organized such that all the decode lookup values for each kernel are grouped together.

If the <number of decode lookup offset records> is greater than 1, the data for each kernel is organized into tables. Typically, the tables represent the lookup values for each row of the kernel. The <number of decode lookup records> and the <number of values per decode lookup record> can be used to determine the structure of the code book when the <number of decode lookup tables> is greater than 1.

B.3.3.4 Spatial data section

The spatial data section of the BIIF image data section is organized such that several different file formats (IMODEs), including band interleaved by pixel, band sequential, and band interleaved by block can be accommodated. In addition, the spatial data subsection is partitioned into one or more image block tables (or subframe tables). In all, there are 5 levels of organization above the /image code/ values. Current implementation of VQ within BIIF uses a single band with an associated LUT; therefore IMODE is B, or band interleaved by block.

The [encoded image data] section of the vector quantized file is comprised of:

- the [spectral group] organization, present in the VQ BIIF image data section to allow for the inclusion of multispectral images that are blocked, but are represented as a band sequential image, limited to 1:
- the image, organized into one or more image blocks, each of which is contained in a [subframe table], defined by the number of blocks per row (NBPR) field of the BIIF image subheader and the number of blocks per column (NBPC) field of the BIIF image subheader and identified by the number of [image block tables] in the spatial data subsection;
- one or more [spectral band tables], which define how the pixels are organized;
- the [image row] level of organization corresponding to the <number of image rows> in the VQ header data;
- the [spectral band line] level of organization, corresponding to the <number of image code/s per row> in the [image display parameter subheader].

B.4 File organization

Fields containing identification and origination information, file security information, and the number and size of the data items contained in the BIIF fields are located in the BIIF file header. Information required to decode the file is located in the image subheader and the BIIF VQ image data section. Within the image data section, multi-byte fields are written in the big endian format. Figure B.6 is a field-by-field description of the BIIF image data section, as used for a vector quantized file. The mask subsection (shown in schematic on the Figure B.2) is shown at a high level only. The specific fields and definitions for the mask subsection are provided in the BIIF base standard.

```
{1}
[BIIF image data]
        {2}
        <blocked image data offset>,uint:4(0,1)
        [mask subsection] (0,1)
                {3}
                [mask subheader]
                [block mask table] (0,1)
                [transparency mask table] (0,1)
        {2}
        [VQ Header]
                {3}
                [image display parameter sub-header]
                        {4}
                        <number of image rows>, uint:4
                        <number of image codes per row>,uint:4
                        <image code bit length>,uint:1
                {3}
                [decode section] (0,1)
                        {4}
                        [decode section subheader]
                                {5}
                                <decode algorithm id>,uint:2
                                <number of decode lookup offset records>, uint:2
                                <number of decode parameter offset records>,uint:2
                                        {6}
                                        [decode lookup offset record] (1, ....many)
                        {4}
                        [decode lookup subsection] (0,1)
                                {5}
                                <decode lookup offset table offset>, uint:2
                                <decode lookup table offset record length>, uint:2
                                [decode offset lookup table]
                                                {7}
                                                <decode lookup table id>, uint:2
                                                <number of decode lookup records>, uint:4
                                                <number of values per decode lookup record>,uint:2
                                                <decode lookup value bit length>, uint:2
                                                <decode lookup table offset>,uint:4
                                {5}
                                [decode lookup table] (1, ...many)
                                        {6}
                                        [decode lookup record] (1,... many)
                                                {7}
                                                /decode lookup value/, bits:var (1,  ... many)
        {2}
        [encoded image data]
                {3}
                [spectral group] (1, ...many)
                        {4}
                        [subframe table] (1, ....many)
                                {5}
                                [spectral band table] (1, ....many)
                                        {6}
                                        [image row] (1, ... many)
                                                {7}
                                                [spectral band line] (1, ...many)
                                                        {8}
                                                        /image code/,bits:var(1, ... many)
```

Figure B.6 -- Structure of the BIIF VQ image data section

B.5 Definitions - image data section

This subclause lists the elements of the VQ header and quantized image data sections, listed in alphabetical order.

1. (<blocked image data offset> ::= a 4-byte unsigned integer defining the offset in bytes of the [encoded image data] from the beginning of the [BIIF image data] section (labeled "image data" in Figure B.2). This field is present only for masked images.
2. <decode algorithm id> ::= a 2-byte unsigned integer defining the algorithm used for the image data in the [frame file]. ::= 1 to indicate that this image data is vector quantized.
3. <decode lookup offset table offset> ::= a 4-byte unsigned integer indicating the displacement, measure in bytes, between the beginning of the [decode lookup subsection] and the first byte of the decode lookup offset table]. The first byte of the [decode lookup subsection] is counted as 0.
4. <decode lookup table id> ::= a 2-byte unsigned integer identifying the [lookup table] described in this [decode lookup offset record], encoded as follows

    a) = 1 to indicate this is row 0 of a 4 x 4 kernel,
    b) = 2 to indicate this is row 1 of a 4 x 4 kernel,
    c) = 3 to indicate this is row 2 of a 4 x 4 kernel,
    d) = 4 to indicate this is row 3 of a 4 x 4 kernel,
    e) = 5 to indicate this is a 16-element, 4 x 4 kernel,
    f) = 6 to indicate this is a 4-element, 2 x 2 kernel.

The nth[decode lookup offset record] shall contain the <decode lookup table id> of the nth [decode lookup table] in this [decode lookup subsection].

5. <decode lookup table offset> ::= a 4-byte unsigned integer defining the displacement, measured in bytes, between the beginning of the [decode lookup subsection] and the first byte of the [decode lookup table] identified in this [decode lookup offset record]. The first byte of the [decode lookup subsection] is counted as 0.
6. <decode lookup table offset record length> ::= a 2-byte unsigned integer indicating the length of each [decode lookup offset record].
7. /decode lookup value/ ::- a variable-length bit field specifying a value in the VQ code book. For a particular VQ scheme, the /decode lookup value/ shall have affixed length, which is defined in the <decode lookup value bit length>.
8. <decode lookup value bit length> ::= a 2-byte unsigned integer >= 4, defining the length in bits of the /decode lookup value/ field in each [decode lookup record] of each [decode lookup table] in the [decode section]. All /decode lookup value/ fields in a given [decode lookup table] shall have the same <decode lookup value bit length>, which shall be a multiple of 4 bits.
9. /image code/ ::= a variable-length bit string indicating an index to the associated VQ code book in a vector quantized map or image file. Successive /image code/ values in a given [image row] shall be a multiple of 8 bits, to ensure that each [image row] consists of an integer number of bytes.
10. <image code bit length> ::= a 1-byte unsigned integer defining the length, in bit, of /image code/.
11. <number of decode lookup records> ::= a 4-byte unsigned integer >= 1, indicating the number of [decode lookup record]s in each [decode lookup table].
12. <number of decode lookup offset records> ::= a 2-byte unsigned integer ^{3} 1, indicating the number of [decode lookup offset record]s in the [decode lookup offset table].
13. <number of decode parameter offset records> ::= a 2-byte unsigned integer >=0, indicating the number of [decode parameter offset record]s in the [decode parameter subsection]. For VQ images, no [decode parameter offset record] is present and therefore, this value shall ::= 0.
14. <number of image codes per row> ::= a 4-byte unsigned integer >= 1, defining, the number of /image code/fields in each [image row] of each [color band table]. All [image row]s in every [spectral band table] in every [subframe table] shall contain the same number of contiguous /image code/s. The <number of image codes per row> shall be chosen to ensure that the total number of bits in the/image code/s constituting a give [image row] shall be a multiple of 8 bits, to ensure that each [image row] consists of an integer number of bytes.
15. <number of image rows> ::= a 4-byte unsigned integer >= 1, indicating the number of [image row]s in each [spectral band table]. All [spectral band table]s in every [subframe table] shall contain the same number of [image row]s.
16. <number of values per decode lookup record> ::= a 2-byte unsigned integer >= 1, indicating the number of contiguous /decode lookup value/ fields in each [decode lookup record] of a given [decode lookup table]. All [decode lookup table]s in a given [decode section] shall have the same number of /decode lookup value/fields in each [decode lookup record].

B.6 Definitions - BIIF header and image subheader

Table B.1 provides specific data values for BIIF header and image subheader fields particular to VQ data.

Table B.1 -- BIIF header and subheader specified data values

FIELD	NAME	SIZE	VALUE RANGE	TYPE
IC	Image Construct. Specific values are identified for masked and unmasked vector quantized images.	2	For VQ images: C4 = VQ image, not masked M4 = VQ image, masked	R
PVTYPE	Pixel Value Type. Type of computer representation used for the value of each /image code/ in the BIIF image.	3	For VQ images: INT = integer	R
CLEVEL	Complexity Level. This field shall contain the complexity level required to interpret fully all components of the file. Valid entries are integers assigned in accordance with complexity requirements established in ANNEX C of this Appendix.	2	For this model profile, applicable CLEVELS are: 01 = Monochromatic BIIF VQ files ² 1024 x 1024 in size with no blocking. 02 = Blocked and/or color-mapped or multispectral, or VQ images larger than 1024 x 1024.	R

Annex C (normative)
Profiling BIIF

C.1 Profiling process

This Annex provide tables that are to be filled in to define a selective application of BIIF. Completion of all the tables below and supporting documents for special definitions constitute a "profile" of BIIF. (In ISO functional profiling taxonomy, this type of profile is referred to as a "format profile" since it defines a data format for use in interchange or sharing of information.)

Profiling begins by identifying the BIIF file parts that are required to support the application and the fields within each file part that are affected. Tables C.1 through C.8 correspond to Tables 1-8 in Subclause 4. The profile definer must derive value ranges for each affected field in each supported part and provide these values in the tables provided in the Data Conformance Specification clause.

C.2 Profile proforma

C.2.1 Use of the model profile

The BIIF Model Profile is provided as a profile which is considered a minimally conformant usage of BIIF. As indicated in instructions below, the profile designer may elect to accept the Model Profile value for each part and field simply by indicating "Same as Model Profile" in the appropriate column.

A Profile Variant (PV) set for each file part is included in the tables of this clause. PVV fields have a complete syntactic and semantic definition in this standard (in clause 4), but typically may have further restriction of value ranges for a particular application. The Model Profile also addresses use of PVU fields by defining structure and representative values for the fields. As such, the model profile provides a recommended starting point for other profiles.

C.2.2 Rules for filling out the proforma tables

All tables below are to be completed in accordance with the following rules and recommendations. Developers are encouraged to accept the Model Profile definitions for all PVU fields, and given the opportunity, where appropriate, to accept the Model Profile constraints for PVV fields. The rules below (designated R1 through R11) shall be followed in filling out the proforma tables for any profile that is to be nominated for registration and/or progression as an ISP.

R1 An expression of value range or a list of allowable BCS entries and a definition of the significance of each entry. When no list or range is present, the default will be to fill the field with spaces (0x20) for BCS-A fields and zeros (0x 30) for BCS-N fields.

R2 The profile definition may constrain allowable field entries to further define the allowable syntax in a manner similar to that of the date and time fields, e.g. CCYYMMDDhhmmss.

R3 The profile definition may define sub-fields. The definition shall be done in a similar manner as to that used in clause 4 for defining fields in this standard. A separately documented definition may be referenced in the proforma and attached to the proforma.

R4 The profile definition for security-related fields shall be identical for all header and subheaders (i.e. FSEC, ISCSEC, SSSEC, TSSEC, etc.). A separately documented definition may be referenced in the proforma and attached to the proforma.

R5 The profile may indicate a numeric value range for the field in the form of Minimum-value - Maximum-value (inclusive) specification. The values may be integers, or in some cases other numeric types as applicable to the fields defined in Clause 4.

R6 Where indicated, a unique string identifier shall be provided. The domain across which uniqueness must be maintained is further clarified as follows: 1) within a BIIF file instance, 2) within profile, 3) within ISO standards, and within an authorized registry.

R7 For some fields there must be a unique NULL/ZERO/NONE indication because of a conditional field dependency on this value. This indicator is usually "N" and shall appear first in the enumerated set of values, for example: {"N", "KEY1", "KEY2"}

R8 Some fields are designated to contain a textual comment (general information, name, etc.) that may be language specific but is otherwise unstructured.

R10 For PVV designated fields, the profile definer may indicate the value constraint on the field as follows:

a) Any value permitted by the standard (as specified in Clause 4);
b) A specific set of values or single value from an enumerated set (Rule 1) or a range of values (Rule 5) as permitted by the standard; or,
c) The Model Profile constraint (by indicating “Same as Model Profile”).

R11 For PVU designated fields, the profile definer is required to provide a complete discussion of the field use, including the following points:

a) Any substructuring of the field (Rule 3);
b) Any syntactic constraint against either the entire field and/or sub-fields (Rule 2); and,
c) An explanation of the meaning of the field and its subfields, including detailed examples of usage.

C.2.2.3 Profile Tables

The following clauses provide the proforma (tables) for profile specification. They also specify the model profile of BIIF.

C.2.2.3.1 BIIF File Header

Table C.1 is the proforma for profile specification of BIIF file header fields.

Table C.1 -- File header fields

FIELD	TYPE	CE / SIZE	PROFILE OPTIONS & RULES	MODEL PROFILE	NEW PROFILE SPECIFICATION Profile Name:_____
			Specify whether this new profile specification includes the entire model profile BFMP as a capability in addition to the new profile capabilities (Clause 5.1).		__ Model profile fully included (can produce and/or read both BFMP files and the new profile specification files) __ Model profile not fully included (can only produce and/or read the new profile specification files)
FHDR	R	A/4	Unique profile name not already registered.	BFMP	__ __ __ __
FVER	R	A/5	Version identifier unique to the registered profile name.	01.00	__ __ __ __ __
CLEVEL	R	A/2	A value of 00 indicates the profile has no internal hierarchy. If an internal hierarchy is desired, the profile may designate two or more values in the range 01-99 to differentiate parameters, values, and ranges within the internal hierarchy. Each profile entry shall specify the CLEVEL hierarchy constraints where applicable. When profile entries do not specify CLEVEL constraints, they are applicable across all CLEVELs. The profile entry shall also define the extent of the Common Coordinate System (CCS) applicable for each designated CLEVEL.	01 02 The CCS extent for each CLEVEL is: 01 1024x1024 02 2048x2048	Select ONE of the following: ___ Same as Model Profile ___ CLEVEL=00 The CCS extent=_____ ___ List allowed values and CCS extent below. Enter CLEVEL specification for each applicable field in the profile. List:
STYPE	R	A/4	Always BF01for all profiles of this standard.	BF01	_X_ Same as Model Profile

Table C.1 -- File header fields (continued)

FIELD	TYPE	CE / SIZE	PROFILE OPTIONS & RULES	MODEL PROFILE	NEW PROFILE SPECIFICATION Profile Name:_____
OSTAID	R	A/10	Any BCS-A string is allowed. Profile may specify further constraints.	Any BCS-A string.	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints:
FDT	R	N/14	As specified in Table 1 for all profiles.	CCYYMMDDhhmmss	_X_ Same as Model Profile
FTITLE	R	U8/80	Any UTF-8 string is allowed. Profile may specify further constraints.	Any BCS-A string.	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints:
FSEC	R	U8/167	The profile definition for security-related fields shall be identical for all header and subheaders (i.e., FSEC, ISCSEC, SSSEC, TSSEC, etc.). The profile definition may define subfields and associated constraints using Table C.2.	See Table C.2	Select ONE of the following: ___ Same as Model Profile ___ The profile specific constraints are as specified in the new profile specification of Table C.2.
FSCOP	R/PVU	5	Any BCS-N string is allowed. Profile may specify further constraints.	00000-99999 An entry of 00000 shall mean that no count of copies is being maintained.	Select ONE of the following: ___ Same as Model Profile ___ The profile specific constraints are as specified in Table C.2.
FSCPYS	R/PVU	5	Any BCS-N string is allowed. Profile may specify further constraints.	00000-99999 An entry of 00000 shall mean that no total file count is being tracked.	Select ONE of the following: ___ Same as Model Profile ___ The profile specific constraints are as specified in Table C.2.
ENCRYP	R/PVU	A/1	Any BCS-A character codes are allowed. Profile shall define the meaning of each code.	0 = not encrypted.	

Table C.1 -- File header fields *(continued)*

FIELD	TYPE	CE / SIZE	PROFILE OPTIONS & RULES	MODEL PROFILE	NEW PROFILE SPECIFICATION Profile Name:_____
OID	R	A/45	Any BCS-A string is allowed. Profile may define subfields and specify associated constraints.	Any BCS-A string. No subfields defined.	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints:
FL	R	N/12	As specified in Table 1 for all profiles.	Calculated value.	_X_ Same as Model Profile
HL	R	N/6	As specified in Table 1 for all profiles.	Calculated value.	_X_ Same as Model Profile
NUMI	R	N/3	Profile shall specify the number or range of image segments allowed in a BIIF file conforming to the profile.	CLEVEL 01: 000-001 CLEVEL 02: 000-020	Select ONE of the following: ___ Same as Model Profile ___ The profile specified range is:
LISHn & LIn	C C	N/6 N/10	A pair of length values as specified in Table 1. These fields repeat (in pairs) the number of times identified by NUMI.	Calculated.	_X_ Same as Model Profile
NUMS	R	N/3	Profile shall specify the number or range of symbol segments allowed in a BIIF file conforming to the profile.	CLEVEL 01: 000 CLEVEL 02: 000-100	Select ONE of the following: ___ Same as Model Profile ___ The profile specified range is:
LSSHn & LSn	C C	N/4 N/6	A pair of length values as specified in Table 1. The fields repeat (in pairs) the number of times identified by NUMS.	Calculated.	_X_ Same as Model Profile
NUMX	R	N/3	Always 000 for all profiles.	000	_X_ Same as Model Profile
NUMT	R	N/3	Profile shall specify the number or range of text segments allowed in a BIIF file conforming to the profile.	CLEVEL 01: 000 CLEVEL 02: 000-010	Select ONE of the following: ___ Same as Model Profile ___ The profile specified range is:
LTSHn & LTn	C C	N/4 N/5	A pair of length values as specified in Table 1. The fields repeat (in pairs) the number of times identified by NUMT.	Calculated.	_X_ Same as Model Profile

Table C.1 -- File header fields *(continued)*

FIELD	TYPE	CE / SIZE	PROFILE OPTIONS & RULES	MODEL PROFILE	NEW PROFILE SPECIFICATION Profile Name:_____
NUMDES	R	N/3	Profile shall specify the number or range of data extension segments allowed in a BIIF file conforming to the profile. When DESs are allowed, the profile shall identify the specific DESs supported at the time of profile registration. Additional DES types can be added to the profile through registration process.	CLEVEL 01: 000 CLEVEL 02: 000-020 Supported DESs: DESTAG: TRE_OVERFLOW DESTAG: TRANSPORTABLE_FILE_STRUCT	Select ONE of the following: ___ Same as Model Profile ___ The profile specified range & list is:
LDSHn & LDn	C C	N/4 N/9	A pair of length values as specified in Table 1. The fields repeat in pairs the number of times identified by NUMDES.	Calculated.	_X_ Same as Model Profile
NUMRES	R	N/3	Profile shall specify the number or range of registered extension segments allowed in a BIIF file conforming to the profile. When RESs are allowed, the profile shall identify the specific DESs supported at the time of profile registration. Additional RES types may be added to the profile through the registration process.	CLEVEL 01: 000 CLEVEL 02: 000 Supported RESs: None	Select ONE of the following: ___ Same as Model Profile ___ The profile specified range is:
LRSHn & LRn	C C	N/4 N/7	A pair of length values as specified in Table 1. The fields repeat in pairs the number of times identified by NUMRES.	Calculated.	_X_ Same as Model Profile
UDHDL	R	N/5	As specified in Table 1 for all profiles.	Calculated.	_X_ Same as Model Profile
UDHOFL	C	N/3	As specified in Table 1 for all profiles.	Calculated.	_X_ Same as Model Profile

Table C.1 -- File header fields *(concluded)*

FIELD	TYPE	CE / SIZE	PROFILE OPTIONS & RULES	MODEL PROFILE	NEW PROFILE SPECIFICATION Profile Name:_____
UDHD	C		As specified in Table 1 for all profiles.	Any tagged record extension general to the BIIF file.	___ Same as Model Profile Select all that apply: ___ TREs are prohibited ___ All Public TREs are allowed ___ All Private TREs are allowed ___ Specific Public TREs are allowed List allowed TREs: ___ Specific Private TREs are allowed List allowed TREs:
XHDL	R	N/5	As specified in Table 1 for all profiles.	Calculated.	_X_ Same as Model Profile
XHDLOFL	C	N/3	As specified in Table 1 for all profiles.	Calculated.	_X_ Same as Model Profile
XHD	C		As specified in Table 1 for all profiles.	Any tagged record extension general to the BIIF file.	___ Same as Model Profile Select all that apply: ___ TREs are prohibited ___ All Public TREs are allowed ___ All Private TREs are allowed ___ Specific Public TREs are allowed List allowed TREs: ___ Specific Private TREs are allowed List allowed TREs:

C.2.2.3.2 BIIF security fields

Table C.2 is the proforma for profile specification of BIIF security fields.

Table C.2 -- Security fields specification

FIELD	TYPE	CE/ SIZE	PROFILE OPTIONS & RULES	MODEL PROFILE	NEW PROFILE SPECIFICATION Profile Name:_____
FSEC ISCSEC SSSEC TSSEC DESCLAS RESCLAS	R R R R R R	U8/167 U8/167 U8/167 U8/167 U8/167 U8/167	The profile definition for security-related fields shall be identical for all header and subheaders (FSEC, ISCSEC, SSSEC, TSSEC, DESCLAS, RESCLAS). The profile definition may define subfields and associated constraints using this table.	The model profile specification for subfields is as described below.	Select ONE of the following: ___ Same as Model Profile ___ The profile specific definition and constraint are specified below.
MODEL PROFILE SPECIFICATION					
SUBFIELD	**TYPE**	**CE/ SIZE**	**DESCRIPTION**	**VALUE RANGE**	**NOTES**
SEC1	R	A/1	Security code applicable to the entire file. The only defined code for the model profile is: U - Unspecified.	U	The model profile does not detail specific file security parameters.
SEC2	R	U8/166	Free form text. Any set of BCS-A characters is allowed. Default entry is all BCS-A spaces.	Any BCS-A. Default is all BCS-A spaces.	
NEW PROFILE SPECIFICATION					
SUBFIELD	**TYPE**	**CE/ SIZE**	**DESCRIPTION**	**VALUE RANGE**	**NOTES**

C.2.2.3.3 BIIF image subheader fields

Table C.3 is the proforma for profile specification of BIIF image subheader fields.

Table C.3 -- Image subheader fields

FIELD	TYPE	CE/ SIZE	PROFILE OPTIONS & RULES	MODEL PROFILE	NEW PROFILE SPECIFICATION Profile Name:_____
IM	R	A/2	Always IM for all profiles.	IM	_X_ Same as Model Profile
IID	R	A/10	Any BCS-A string is allowed. Profile may specify further constraints.	Any BCS-A string.	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints:
IDATIM	R	N/14	As specified in Table 2 for all profiles.	CCYYMMDDhhmmss.	_X_ Same as Model Profile
IINFO	R	U8/97	Any UTF-8 string is allowed. The profile definition may define subfields and associated constraints.	Any BCS-A string.	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints:
ISCSEC	R	U8/167	As specified in Table C.2.	As specified in Table C.2.	As specified in Table C.2
ENCRYP	R/PVU	A/1	Any BCS-A character code is allowed. Profile shall define the meaning of each code.	0 = not encrypted.	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints:
ISORCE	R	U8/42	Any BCS-A string is allowed. The profile definition may define subfields and associated constraints.	Any BCS-A string.	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints:
NROWS	R	N/8	Profile shall specify the range for the number of rows of pixel values allowed in a BIIF image conforming to the profile.	CLEVEL 01: 00000001-00001024 CLEVEL 02: 00000001-00002048	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints:

Table C.3 -- Image subheader fields *(continued)*

FIELD	TYPE	CE/ SIZE	PROFILE OPTIONS & RULES	MODEL PROFILE	NEW PROFILE SPECIFICATION Profile Name:_____
NCOLS	R	N/8	Profile shall specify the range for the number of columns of pixel values allowed in a BIIF file conforming to the profile.	CLEVEL 01: 00000001-00001024 CLEVEL 02: 00000001-00002048	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints:
PVTYPE	R	A/3	Select from the following allowed values: B INT SI R C	CLEVEL 01: INT CLEVEL 02: B INT	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints:
IREP	R	A/8	Select from the following values: MONO RGB RGB/LUT HIS CMY CMYK YIQ YUV YCbCr CIE 1D 2D ND MULTI PIKS ----and/or---- Specify additional values and their specified use.	CLEVEL 01: MONO PIKS CLEVEL 02: MONO RGB RGB/LUT YCbCr PIKS	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints:

Table C.3 -- Image subheader fields *(continued)*

FIELD	TYPE	CE / SIZE	PROFILE OPTIONS & RULES	MODEL PROFILE	NEW PROFILE SPECIFICATION Profile Name:_____
ICAT	R	A/8	Select from the following values: VIS - Visual imagery SL - Side looking radar TI - Thermal infrared FL - Forward looking IR RD - Radar EO - Electro-optical OP - Optical HR - High res. radar HS - Hyperspectral CP - Color frame photo BP - Black/white photo SAR - Synthetic Aperature Radar SARIQ - SAR radio hologram IR - Infrared MS - Multi-spectral FP - Finger prints MRI - Magnetic Resonance imagery XRAY - x-rays CAT - CAT scan MAP - Raster map PAT - Colour patch LEG - Legend DTEM - Elevation model data MATR - general matrix data LOCG - Location grids ----and/or---- Specify additional values and their specified use.	CLEVEL 01: VIS CLEVEL 02: All listed values for ICAT in Table 3.	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints:
ABPP	R	N/2	Select from the following options: 01 04 08 11 12 01-08 09-16 17-32 33-64 65-96	CLEVEL 01: 08 CLEVEL 02: 01 08 11 12	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints:

Table C.3 -- Image subheader fields *(continued)*

FIELD	TYPE	CE / SIZE	PROFILE OPTIONS & RULES	MODEL PROFILE	NEW PROFILE SPECIFICATION Profile Name:_____
PJUST	R	A/1	Select from the following options: R L	R	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints:
ICORDS	R	A/1	Space character - or - Specify and define codes	CLEVEL 01: Space CLEVEL 02: Space	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints:
IGEOLO	C	A/60	For each code specified in ICORDS, define the content and structure of this field.	Omitted.	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints:
NICOM	R	N/1	Select any range within 0-9.	0-9	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints:
ICOMn	C	U8/80	Any UTF-8 string is allowed. Profile may specify further constraints.	Any UTF-8 string.	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints:

Table C.3 -- Image subheader fields *(continued)*

FIELD	TYPE	BCS / SIZE	PROFILE OPTIONS & RULES	MODEL PROFILE	NEW PROFILE SPECIFICATION Profile Name:_____
IC	R	A/2	Any BCS-A string is allowed to identify the type of compression used in the image data field. The character M is reserved for use to indicate that pixel mask(s) have been included in the image data field. Reference must be made to the specification document describing the compression. Representative values are: NC - uncompressed NM - uncompressed with pixel mask tables C1 - C9 M1 - M9	CLEVEL 01: NC - Uncompressed CLEVEL 02: NC- Uncompressed NM - Uncompressed with mask table(s). C1/M1 - Bi-Tonal compression per ITU-T T.4, AMD2 08/95 C3/M3 - JPEG lossy DCT compression per ISO/IEC 10918-1 and ISO/IEC 10918-3. C4/M4 - VQ (Annex B) C5/M5 - JPEG lossless compression per ISO/IEC 10918-1 and ISO/IEC 10918-3.	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints:

Table C.3 -- Image subheader fields *(continued)*

FIELD	TYPE	BCS / SIZE	PROFILE OPTIONS & RULES	MODEL PROFILE	NEW PROFILE SPECIFICATION Profile Name:_____
COMRAT	C	A/4	The field is omitted when IC=NC or IC=NM. When present, the profile shall describe the use of the field for describing compression rate related information.	This conditional field is only present when the IC code is other than NC or NM. ELSE: For C1/M1: 1D - one dimensional coding 2DS- two dimensional coding, standard vertical resolution (K=2) 2DH- two dimensional coding, high vertical resolution (K=4) See ITU-T T.4, AMD2 08/95 For C3/M3 the value is always 00.0. For C4/M4 the value is in the form of nn.n representing the approximate number of bits per pixel for the compressed image.	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints:
NBANDS	R	N/1	Identify allowed values from: 0 1-9 T	CLEVEL 01: 1 CLEVEL 02: 1 3 T	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints:
XBANDS	C	N/5	Conditional field; omitted unless NBANDS=0.	CLEVEL 01: Not Used. CLEVEL 02: Not Used.	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints:
NOTE: The fields IREPBANDn through LUTnm repeat the number of times indicated in the NBANDS field or the XBANDS field.					

Table C.3 -- Image subheader fields *(continued)*

FIELD	TYPE	BCS / SIZE	PROFILE OPTIONS & RULES	MODEL PROFILE	NEW PROFILE SPECIFICATION Profile Name:_____
IREPBANDn	R	A/2	Identify allowed value(s) for each band which further identifies the significance of the band as related to the value in the IREP field.	For IREP=MONO, two spaces. For IREP=RGB, R, G, B. For IREP=RGB/LUT two spaces. For IREP=YCbCr Y, Cb, Cr. For IREP=PIKS two spaces.	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints:
ISUBCATn	R	A/6	Identify allowed value(s) for each band which further identifies the significance of the band as related to the value in the ICAT field.	For all values of ICAT, the value is six spaces.	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints:
IFCn	R	A/1	Identify filter condition codes and the specification of the corresponding filter condition. The code N means there is no filter condition.	N	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints:
IMFLTn	R	A/3	Identify filter codes corresponding to each filter condition identified in IFCnn. If none, value is three spaces.	Three spaces.	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints:

Table C.3 -- Image subheader fields *(continued)*

FIELD	TYPE	BCS / SIZE	PROFILE OPTIONS & RULES	MODEL PROFILE	NEW PROFILE SPECIFICATION Profile Name:_____
NLUTSn	R	N/1	Identify the allowed range for specifying the number of look up tables (8-bit entries in each table) associated with the band. Typical values are: 0 - no LUTS. 1 - for translating NELUTS number of values to alternate 8-bit (or less) values. 2 - for translating NELUTS number of values into alternate values of 16 or less bits. 3 - For translating NELUTS number of values into alternate values of 24 or less bits.	CLEVEL 01: 0 CLEVEL 02: 1-3	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints:
NELUTn	C	N/5	When NLUTnn=00000, this field is omitted. Otherwise it specifies the number of 8-bit entries in each sequential LUT.	CLEVEL 01: Omitted. CLEVEL 02: Omitted when NLUTSn=0 00001-32768	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints:
LUTDn1	C	Derived from Value NELUT1	This field shall be omitted if the n^{th} Band Number of LUTS is zero. Otherwise, this field shall contain the data defining the first look-up table for the nth image band. Multiple LUTs may be used to translate the index value into multiple octet values.	Data only	
LUTDnm	C	Derived from Value NELUT1	This field shall be omitted if the n^{th} Band Number of LUTs is zero. Otherwise, this field shall contain the data defining the m^{th} look-up table for the n^{th} image band. Each entry in the look-up table is composed of one octet, ordered from most significant bit to least significant bit representing a value from 0 to 255.	Data only	

Table C.3 -- Image subheader fields *(continued)*

FIELD	TYPE	BCS / SIZE	PROFILE OPTIONS & RULES	MODEL PROFILE	NEW PROFILE SPECIFICATION Profile Name:_____
ISYNC	R	A/1	Identify indicator codes for each allowed end of row or end of column marker to be used. For each listed code, identify the specification for the sync code marker. The value 0 indicates no sync code is used.	0	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints:
IMODE	R	A/1	IMODE codes are: B=Block interleaved R=Row interleaved P=Pixel Interleaved S=Band sequential	CLEVEL 01: B CLEVEL 02: B P R S	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints
NBPR	R	N/4	Identify the allowed range.	CLEVEL 01: 0001 CLEVEL 02: 0001 - 0064	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints:
NBPC	R	N/4	Identify the allowed range.	CLEVEL 01: 0001 CLEVEL 02: 0001 - 0064	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints:
NPPBH	R	N/4	Identify the allowed range.	CLEVEL 01: 00032 - 1024 CLEVEL 02: 00032-2048	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints:
NPPBV	R	N/4	Identify the allowed range.	CLEVEL 01: 00032 - 1024 CLEVEL 02: 00032-2048	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints:

Table C.3 -- Image subheader fields *(concluded)*

FIELD	TYPE	BCS / SIZE	PROFILE OPTIONS & RULES	MODEL PROFILE	NEW PROFILE SPECIFICATION Profile Name:_____
NBPP	R	N/2	Select the allowed values from the following: 01 04 08 16 24 32 40 48 56 64 72 80 88 96 -- and/or -- specify additional values and than specify use	CLEVEL 01: 01 CLEVEL 02: 01 08 16	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints:
IDLVL	R	N/3	001-999	001-999	_X_ Same as Model Profile
IALVL	R	N/3	000-998	000-998	_X_ Same as Model Profile
ILOC	R	N/10	As specified in Table 3.	As specified in Table 3.	_X_ Same as Model Profile
IMAG	R	A/4	As specified in Table 3.	As specified in Table 3.	_X_ Same as Model Profile
UDIDL	R	N/5	As specified in Table 3 for all profiles.	Calculated.	_X_ Same as Model Profile
UDOFL	C	N/3	As specified in Table 3 for all profiles.	Calculated.	_X_ Same as Model Profile
UDID	C		As specified in Table 3 for all profiles.	Any tagged record extension specific to the image.	__ Same as Model Profile Select all that apply: ___ TRES are prohibited ___ Any Public TRE ___ Any Private TRE ___ Allowed Public TREs List: ___ Allowed Private TREs Specify constraints
IXSHDL	R	N/5	As specified in Table 3 for all profiles.	Calculated.	_X_ Same as Model Profile
IXSOFL	C	N/3	As specified in Table 3 for all profiles.	Calculated.	_X_ Same as Model Profile
IXSHD	C		As specified in Table 3 for all profiles.	Any tagged extension specific to the image.	_X_ Same as Model Profile

C.2.2.3.4 BIIF image data mask table

Table C.4 is the proforma for profile specification of BIIF image data mask table fields.

Table C.4 -- Image data mask table

FIELD	TYPE	BCS/ SIZE	PROFILE OPTIONS & RULES	MODEL PROFILE	NEW PROFILE SPECIFICATION Profile Name:_____
When the profile specifies use of the image data mask table, the following fields all become applicable. See Table C.3, field, IC. When the image data mask table is not specified in the profile, none of the following fields apply.			When selected codes for the IC field include the character M, the specification of image data mask tables becomes required. Otherwise use of the data mask table is optional.	CLEVEL 01: Image data mask tables not used CLEVEL 02: Used as specified in Table 4.	Select ONE of the following: ___ Same as Model Profile ___ NO. Image data mask tables are not allowed for use within this profile.
IMDATOFF	C	N/4	As specified in Table 4.	As specified in Table 4.	_X_ Same as Model Profile
BMRLNTH	C	N/2	As specified in Table 4.	As specified in Table 4.	_X_ Same as Model Profile
TMRLNTH	C	N/2	As specified in Table 4.	As specified in Table 4.	_X_ Same as Model Profile
TPXCDLNTH	C	N/2	As specified in Table 4.	As specified in Table 4.	_X_ Same as Model Profile
TPXCD	C		As specified in Table 4.	As specified in Table 4.	_X_ Same as Model Profile
BMRnBNDm	C	N/4	As specified in Table 4.	As specified in Table 4.	_X_ Same as Model Profile
TMRnBNDm	C	N4	As specified in Table 4.	As specified in Table 4.	_X_ Same as Model Profile

C.2.2.3.5 BIIF symbol subheader

Table C.5 is the proforma for profile specification of BIIF symbol subheader fields.

Table C.5 -- Symbol subheader

FIELD	TYPE	BCS/ SIZE	PROFILE OPTIONS & RULES	MODEL PROFILE	NEW PROFILE SPECIFICATION Profile Name:_____
SY	R	A/2	Always SY for all profiles.	SY	_X_ Same as Model Profile
SID	R	A/10	Any BCS-A string is allowed. Profile may specify further constraints.	Any BCS-A string.	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints:
SNAME	R	U8/20	Any UTF-8 string is allowed. Profile may specify further constraints.	Any UTF-8 string.	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints:
SSSEC	R	U8/167	As specified in Table C.2.	As specified in Table C.2.	As specified in Table C.2

Table C.5 -- Symbol subheader *(continued)*

FIELD	TYPE	BCS/ SIZE	PROFILE OPTIONS & RULES	MODEL PROFILE	NEW PROFILE SPECIFICATION Profile Name:_____
ENCRYP	R/PVU	A/1	Any BCS-A character code is allowed. Profile shall define the meaning of each code.	Encryption code and meaning defined by the profile 0 = not encrypted.	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints..
SFMT	R	A/1	This field contains a C indicating the symbol data field contains data structured according to ISO/IEC 8632, Computer Graphics Metafile (CGM). Additional symbol format codes may be added through the graphical item registration process.	C	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints:
SSTRUCT	R	A/13	Any BCS-A string is allowed. The profile definition may define subfields and associated constraints.	0000000000000	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints:
SDLVL	R	N/3	001-999	001-999	_X_ Same as Model Profile
SALVL	R	N/3	000-998	000-998	_X_ Same as Model Profile
SLOC	R	N/10	As specified in Table 5.	As specified in Table 5.	_X_ Same as Model Profile
SLOC2	R	N/10	As specified in Table 5.	0000000000	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints:

Table C.5 -- Symbol subheader *(concluded)*

FIELD	TYPE	BCS/ SIZE	PROFILE OPTIONS & RULES	MODEL PROFILE	NEW PROFILE SPECIFICATION Profile Name:_____
SPARMS	R	A/13	Any BCS-A string is allowed. The profile definition may define subfields and associated constraints.	The 13 bytes are structured into two subfields as follows: **Subfield:** SCOLOR **Name:** Symbol Colour. This subfield contains a C if the symbol data contains colour or M if it is monochrome. **Type:** R **CE/SIZE:** A/1 **Value Range:** C, M **Subfield:** SRES2 **Name:** Reserved for future use. **Type:** R **CE/SIZE:** A/12 **Value Range:** 000000000000	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints:
SXSHDL	R	N/5	As specified in Table 5 for all profiles.	Calculated.	_X_ Same as Model Profile
SXSOFL	C	N/3	As specified in Table 5 for all profiles.	Calculated.	_X_ Same as Model Profile
SXSHD	C		As specified in Table 5 for all profiles.	Any tagged record extension specific to the symbol.	_X_ Same as Model Profile Select all that apply: ___ TREs are prohibited ___ Any Public TRE ___ Any Private TRE ___ Allowed Public TREs List: ___ Allowed Private TREs Specify constraints:

C.2.2.3.6 BIIF text subheader

Table C.6 is the proforma for profile specification of BIIF text subheader fields.

Table C.6 -- Text subheader

FIELD	TYPE	BCS/ SIZE	PROFILE OPTIONS & RULES	MODEL PROFILE	NEW PROFILE SPECIFICATION Profile Name:_____
TE	R	A/2	Always TE for all profiles.	TE	_X_ Same as Model Profile
TEXTID	R	A/10	Any BCS-A string is allowed. Profile may specify further constraints.	Any BCS-A string.	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints:
TXTDT	R	N/14	As specified in Table 6.	As specified in Table 6.	_X_ Same as Model Profile
TXTITL	R	U8/80	Any BCS-A string is allowed. Profile may specify further constraints.	Any BCS-A string.	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints:
TSSEC	R	U8/167	As specified in Table C.2.	As specified in Table C.2.	_X_ Same as Model Profile
TXTFMT	R	A/3	Any BCS-A code is allowed. Profile may specify further constraints. Representative values are: STA - Standard BCS-A. Any BCS-A characters are allowed in the text data field. UC2 - Standard UCS-2 UC4 - Standard UCS-4 UT1 - Standard UTF-1 UT8 - Standard UTF-8	CLEVEL 01: STA CLEVEL 02: STA UC4	Select ONE of the following: ___ Same as Model Profile ___ The following are profile specific constraints:
TXSHDL	R	N/5	As specified in Table 6 for all profiles.	Calculated.	_X_ Same as Model Profile
TXSOFL	C	N/3	As specified in Table 6 for all profiles.	Calculated.	_X_ Same as Model Profile
TXSHD	C		As specified in Table 6 for all profiles.	Any tagged extension specific to the text segment.	_X_ Same as Model Profile Select all that apply: ___ TREs are prohibited ___ Any Public TRE ___ Any Private TRE ___ Allowed Public TREs List: ___ Allowed Private TREs Specify constraints:

C.2.2.3.7 BIIF tagged record extensions (TREs)

The use and registration of BIIF registered extensions is not tied to any specific BIIF profile registration action. A BIIF profile can designate if none, any, or specific registered extensions are allowed within BIIF files conforming to the specific profile. No specific registered extensions are specified for use within the model profile. The model profile allows for the inclusion of any registered extension.

Submission of registered extensions for registration shall as a minimum use the proforma of Table C.7. The submission shall include additional documentation as needed to fully describe the proposed extension.

Table C.7 -- Tagged record extensions

FIELD	TYPE	BCS/ SIZE	OPTIONS & RULES	NAME/DESCRIPTION	VALUE RANGE SPECIFICATION
TRETAG	R	A/6	Unique name not already registered as specified in Table 7.	Unique extension type identifier.	
TREL	R	N/5	As specified in Table 7.	Length of REDATA field.	
TREDATA	R		Extend this table to fully define the contents of the data field. Provide additional narrative if needed to provide a comprehensive description of the extension data.	Extension data.	

C.2.2.3.8 BIIF data extension segments

The use and registration of BIIF data extension segments is not tied to any specific BIIF profile registration action. A BIIF profile can designate whether none, any, or specific data extension segments are allowed within BIIF files conforming to the specific profile. Two specific data extension segments are specified for use within the model profile. They are: TRE_OVERFLOW and TRANSPORTABLE_FILE_STRUCT.

Submission of data extension segments for registration shall as a minimum use the proforma of Table C.8A. The submission shall include additional documentation as needed to fully describe the proposed data extension segment.

Table C.8a -- Data extension segment proforma

FIELD	TYPE	BCS/ SIZE	OPTIONS & RULES	NAME/DESCRIPTION	VALUE RANGE SPECIFICATION
DE	R	A/2	Always DE.	File part type.	DE
DESTAG	R	A/25	Unique BCS-A string not already registered.	Unique DES type identifier.	
DESVER	R	N/2	Version identifier.	Version of the DESTAG.	
DESCLAS	R	A/167	Not defined when registering the DES. The structure and parameters shall be those specified in Table C.2 for the BIIF profile of the file in which the DES appears.	Security specific parameters.	Not defined when registering the DES. The structure and parameters shall be those specified in Table C.2 for the BIIF profile of the file in which the DES appears.
DESOFLW	C	A/6	As specified in Table 8.	Overflowed header type.	When field is present: UDHD UDID XHD IXSHD SXSHD TXSHD
DESITEM	C	N/3	As specified in Table 8.	Data item reference.	000 - 999
DESSHL	R	N/4	As specified in Table 8.	Length of profile defined subheader fields.	0000 - 9999
DESSHF	C	Calc.	Extend this table to fully define each subfield applicable to the DES.	Profile defined subheader fields.	
DESDATA	R	Calc.	Extend this table to fully define the contents of the data field. Provide additional narrative if needed to provide a comprehensive description of the DES.	Profile defined data field.	

C.2.2.3.9 BIIF reserved extension segments

The use and registration of BIIF reserved extension segments is not tied to any specific BIIF profile registration action. A BIIF profile can designate whether none, any, or specific reserved extension segments are allowed within BIIF files conforming to the specific profile. No specific reserved extension segments are specified for use within the model profile.

Submission of reserved extension segments for registration shall as a minimum use the proforma of Table C.8b. The submission shall include additional documentation as needed to fully describe the proposed reserved extension segment.

Table C.8b -- Reserved extension segment proforma

FIELD	TYPE	BCS/ SIZE	OPTIONS & RULES	NAME/DESCRIPTION	VALUE RANGE SPECIFICATION
RE	R	A/2	Always RE.	File part type.	RE
RESTAG	R	A/25	Unique BCS-A string not already registered.	Unique RES type identifier.	
RESVER	R	N/2	Version identifier.	Version of the RESTAG.	
RESCLAS	R	A/167	Not defined when registering the RES. The structure and parameters shall be those specified in Table C.2 for the BIIF profile of the file in which the RES appears.	Security specific parameters.	Not defined when registering the RES. The structure and parameters shall be those specified in Table C.2 for the BIIF profile of the file in which the RES appears.
RESOFLW	C	A/6	As specified in Table 8.	Overflowed header type.	When field is present: UDHD UDID XHD IXSHD SXSHD TXSHD
RESITEM	C	N/3	As specified in Table 8.	Data item reference.	000 - 999
RESSHL	R	N/4	As specified in Table 8.	Length of profile defined subheader fields.	0000 - 9999
RESSHF	C	Calc.	Extend this table to fully define each subfield applicable to the RES.	Profile defined subheader fields.	
RESDATA	R	Calc.	Extend this table to fully define the contents of the data field. Provide additional narrative if needed to provide a comprehensive description of the RES.	Profile defined data field.	

C.2.2.3.10 TFS Profile Proforma

The rules for defining valid profiles for TFS have the following objectives:

1. Provide interoperability between applications by specifically defining constraints on the TFS.
2. Provide a framework for developing TFS profiles.
3. Provide a BIIF TFS model profile for use with a BIIF Model Profile.
4. Set uniform rules for the development of conformance tests.

The BIIF TFS model profile is a usable instance of the TFS. The BIIF TFS model profile is a starting point from which an application-specific TFS profile should be defined. When creating a TFS profile, consideration should be given for each of the BIIF Model Profile specifications and either accept the specification or modify them only when they are not adequate.

The following table defines the BIIF TFS Model Profile and provides a mechanism for an application to define a unique TFS profile for conformance. The first column indicates the entry number for compliance, the second column is the requirement or TFS command stated for compliance. The third column is the BIIF TFS Model Profile. Each condition is checked as Required, Optional, Conditional, or Prohibited. The fourth column is provided for the registration of new TFS profiles. The boxes must be checked appropriately and all restrictions for the given command must be explained.

Table C.9 -- TFS profile proforma table

Table Entry Number	TFS Command or Specification	BIIF TFS Model Profile	New Profile Specification
1	Profile name	BIIF-TFS	Must be unique name not already registered.
2	BEGIN TFS	Required: X Optional: _ Conditional: _ Prohibited: _ Restrictions: Must be only one instance in metafile and be the first metafile command. The BEGIN TFS parameter shall contain the following data, delimited by the "/" character: Profile=profile_name BIIF-MODEL-TFS for model profile, Date=YYYYMMDDhhmmss, and Name="name of TFS". (case insensitive)	Same as model profile: X Profile must include: profile=profile_name, date=YYYYMMDDhhmmss, and name="name of TFS" delimited by the / character (case insensitive):
3	TFS VERSION	Required: X Optional: _ Conditional: _ Prohibited: _ Restrictions: Must be only one instance per metafile and be the second metafile command.	Same as model profile: X
4	TFS SECURITY	Required: X Optional: _ Conditional: _ Prohibited: _ Restrictions: Must be only one instance per metafile and be the third metafile command. The format must match the security used in the BIIF Model Profile.	Same as model profile: _ Required: X Optional: _ Conditional: _ Prohibited: _ Restrictions: The format used for security must match the format used for the BIIF profile.

Table C.9 -- TFS profile proforma table *(continued)*

Table Entry Number	TFS Command or Specification	BIIF TFS Model Profile	New Profile Specification
5	TFS METADATA	Required: X Optional: _ Conditional: _ Prohibited: _ Restrictions: Must describe clearly the contents or purpose of the TFS and its transports contents.	Same as model profile: _ Required: _ Optional: _ Conditional: _ Prohibited: _ Restrictions:
6	TFS INDEX	Required: _ Optional: X Conditional: _ Prohibited: _ Restrictions: When used, it contains each transport name with the byte offset from the beginning of the file to each BEGIN TRANSPORT.	Same as model profile: _ Required: _ Optional: _ Conditional: _ Prohibited: _ Restrictions:
7	TFS SUBSCRIPTION	Required: _ Optional: X Conditional: _ Prohibited: _ Restrictions: Used for requesting TFS to be delivered to user determined by the TFS SUBSCRIPTION parameters. Each object delivery can be defined. Example: XXX=1 defines that object XXX be delivered in the TFS file. There are three subscription options: 1=pull requested data, 2=subscribe to requested data, and 3=cancel subscription.	Same as model profile: _ Required: _ Optional: _ Conditional: _ Prohibited: _ Restrictions:
8	TFS CONFIGURA-TION	Required: _ Optional: X Conditional: X Prohibited: _ Restrictions: Used for providing TFS information or requesting TFS information. When requesting TFS information the TFS CONFIG DATA command must be present.	Same as model profile: _ Required: _ Optional: _ Conditional: _ Prohibited: _ Restrictions:

Table C.9 -- TFS profile proforma table *(continued)*

Table Entry Number	TFS Command or Specification	BIIF TFS Model Profile	New Profile Specification
9	TFS CONFIG DATA	Required: _ Optional: X Conditional: X Prohibited: _ Restrictions: This command is conditionally required when the TFS CONFIGURATION contains 0 to indicate that configuration data is included. This command is optional otherwise. The TFS CONFIGURATION consist of OBJECT_NAME-SERVICE. For example, YYY=HTML. This indicates that an YYY object is of type HTML. It is up to the application to provide the Web Browser to read YYY.	Same as model profile: _ Required: _ Optional: _ Conditional: _ Prohibited: _ Restrictions:
10	TFS Descriptor Commands	Restrictions: All TFS Descriptor Commands shall start after the BEGIN TFS command and end before the BEGIN TRANSPORT command.	Same as model profile: X
11	BEGIN TRANSPORT	Required: _ Optional: X Conditional: X Prohibited: _ Restrictions: Optionally required when one or more profiles are to be transported. The BEGIN TRANSPORT parameter should include the name of the recipient.	Same as model profile: X
12	TRANSPORT SECURITY	Required: _ Optional: _ Conditional: X Prohibited: _ Restrictions: Conditionally required when BEGIN TRANSPORT command is present. Must be only one instance per transport and be the first command after the BEGIN TRANSPORT command. The format must match the security used in the BIIF Model Profile.	Same as model profile: _ Required: _ Optional: _ Conditional: X Prohibited: _ Restrictions: Conditionally required when BEGIN TRANSPORT command is present and be the first metafile command after BEGIN TRANSPORT command. The format used for security must match the format used for the BIIF profile.

Table C.9 -- TFS profile proforma table *(continued)*

Table Entry Number	TFS Command or Specification	BIIF TFS Model Profile	New Profile Specification
13	TRANSPORT METADATA	Required: X Optional: _ Conditional: _ Prohibited: _ Restrictions: Must describe clearly the contents or purpose of the transport and its contents.	Same as model profile: _ Required: _ Optional: _ Conditional: _ Prohibited: _ Restrictions:
14	TRANSPORT INDEX	Required: _ Optional: X Conditional: _ Prohibited: _ Restrictions: When used, it contains each profile name with the byte offset from the beginning of the BEGIN TRANSPORT command to each high level BEGIN PROFILE.	Same as model profile: _ Required: _ Optional: _ Conditional: _ Prohibited: _ Restrictions:
15	Transport Descriptor Commands	Restrictions: All Transport Descriptor Commands for the given transport shall start after the BEGIN TRANSPORT command and end before the BEGIN TRANSPORT BODY command.	Same as model profile: X
16	BEGIN TRANSPORT BODY	Required: _ Optional: _ Conditional: X Prohibited: _ Restrictions: Conditionally required for each BEGIN TRANSPORT command. It must be the first command after the Transport Descriptor Commands.	Same as model profile: X
17	BEGIN PROFILE	Required: _ Optional: X Conditional: X Prohibited: _ Restrictions: Conditionally required when PROFILE ACTION or PROFILE OBJECT commands are present within the given profile state.	Same as model profile: X

Table C.9 -- TFS profile proforma table *(continued)*

Table Entry Number	TFS Command or Specification	BIIF TFS Model Profile	New Profile Specification
18	PROFILE SECURITY	Required: X Optional: _ Conditional: X Prohibited: _ Restrictions: Conditionally required when BEGIN PROFILE command is present. Must be only one instance per profile and be the first command after the BEGIN PROFILE command. The format must match the security used in the BIIF Model Profile.	Same as model profile: _ Required: X Optional: _ Conditional: _ Prohibited: _ Restrictions: The format used for security must match the format used for the BIIF profile.
19	PROFILE METADATA	Required: X Optional: _ Conditional: _ Prohibited: _ Restrictions: Must describe clearly the contents or purpose of the profile and its contents.	Same as model profile: _ Required: _ Optional: _ Conditional: _ Prohibited: _ Restrictions:
20	PROFILE INDEX	Required: _ Optional: X Conditional: _ Prohibited: _ Restrictions: When used, it contains each PROFILE ACTION, PROFILE OBJECT, or an embedded BEGIN PROFILE name with the byte offset from the beginning of the BEGIN PROFILE command to each next level command.	Same as model profile: _ Required: _ Optional: _ Conditional: _ Prohibited: _ Restrictions:
21	BEGIN PROFILE BODY	Required: _ Optional: _ Conditional: X Prohibited: _ Conditionally required for each BEGIN PROFILE command. It must be the first command after the Profile Descriptor Commands.	Same as model profile: X
22	Profile Descriptor Commands	Restrictions: All Profile Descriptor Commands for the given profile shall start after the BEGIN PROFILE command and end before the BEGIN PROFILE BODY command.	Same as model profile: X

Table C.9 -- TFS profile proforma table *(continued)*

Table Entry Number	TFS Command or Specification	BIIF TFS Model Profile	New Profile Specification
23	PROFILE ACTION	Required: _ Optional: X Conditional: _ Prohibited: _ Restrictions: When used parameter must be in the form “OBJECT_NAME= action”. This command must be ignored and documented if not known to the application.	Same as model profile: _ Required: _ Optional: _ Conditional: _ Prohibited: _ Restrictions:
24	PROFILE OBJECT	Required: _ Optional: X Conditional: _ Prohibited: _ Restrictions: Contains OBJECT_TYPE 1or 2 OBJECT_NAME, OBJECT_STORAGE 0 or 1 or 2, and OBJECT_DATA if OBJECT_STORAGE is not 0.	Same as model profile: _ Required: _ Optional: _ Conditional: _ Prohibited: _ Restrictions:
26	END PROFILE	Required: _ Optional: _ Conditional: X Prohibited: _ Restrictions: Must be only one instance per profile state and be the last command in the profile state.	Same as model profile: X
27	END TRANSPORT	Required: _ Optional: _ Conditional: X Prohibited: _ Restrictions: Must be only one instance per transport state and be the last command in the transport state.	Same as model profile: X
28	END TFS	Required: X Optional: _ Conditional: _ Prohibited: _ Restrictions: Must be only one instance per metafile and be the last metafile command.	Same as model profile: X
29	TFS ESCAPE	Required: _ Optional: _ Conditional: _ Prohibited: X Restrictions: All TFS ESCAPE commands are ignored.	Same as model profile: _ Required: _ Optional: _ Conditional: _ Prohibited: _ Restrictions:

Table C.9 -- TFS profile proforma table *(concluded)*

Table Entry Number	TFS Command or Specification	BIIF TFS Model Profile	New Profile Specification
30	Nested Profiles	Restrictions: Nested profiles or profiles within profiles are allowed. Nested levels allowed: 10	Same as model profile: _ Restrictions:
31	Number of Objects	Restrictions: There is no restriction to the number of objects as long as the file sizes of the TFS and BIIF comply within the certification conformance level.	Same as model profile: _ Restrictions:
32	Number of Profiles	Restrictions: There is no restriction to the number of profiles as long as the file sizes of the TFS and BIIF comply within the certification conformance level.	Same as model profile: _ Restrictions:
33	Number of Transports	Restrictions: There is no restriction to the number of transports as long as the file sizes of the TFS and BIIF comply within the certification conformance level.	Same as model profile: _ Restrictions:

ANNEX D (informative)
Implementation Consideration and Product Configurations

D.1 Implementation considerations and product configurations

The Basic Imagery Interchange Format (BIIF) has been developed to provide image exchange capabilities and interoperability among computer systems of various designs and capabilities. For the purposes of BIIF, interoperability means the ability to exchange BIIF formatted imagery products among BIIF capable systems in a manner that is meaningful and useful to the end users. This annex will discuss general considerations pertinent to successful implementation of BIIF. Guidelines will be presented, and potential problems will be highlighted. BIIF preprocessor and postprocessor software, the software potentially necessary to write and read a BIIF file based on host native mode files containing the data items to be included, are to be written by the user. The combination of the preprocessor and postprocessor hereafter will be referred to as the "BIIF implementation." Preprocessing is sometimes called "packing," and postprocessing is called "unpacking."

Subclause D.4 discusses several generalized product configurations that can be used as the basis for defining specific imagery products. These product configurations are typical of those successfully used within the imagery and mapping community to date.

D.2 TRE_OVERFLOW sample

The following is a Sample BIIF File Structure that shows control tags with overflows; the file has a single image.

Table D.1 -- Sample file structure

HEADER	IMAGE SUBHEADER	IMAGE DATA	DATA EXTENSION SUBHEADER	DATA EXTENSION

	MAIN HEADER																	IMAGE SUBHEADER					DES SUBHEADER						
FIELD NAME	FHDR	ETC	FL	HL	NUMI	LISH001	LI001	NUMS	NUMX	NUMT	NUMDES	LDSH001	LD001	NUMRES	UDHDL	XHDL	IM	UIXSHDL	IXSOFL	IXSHD	IMAGE DATA	DE	DESTAG	ETC	DESOFLW	DESITEM	DESSHL	DESDATA	
BYTES	9		1	6	3	6	10	3	3	3	3	4	9	3	5	5	2	5	3	5		2	25		6	3	4	4000	
FIELD VALUE	BFMP01		00008050757 64	000417	001	0984422	00849346 56	000	000	000	001	0249	000042000	000	00000	0000	IM	000000	980003	001	TAG123		DE	TRE_OVERFLOW		UDID	001	0000		
												TAG 1 (32,000 BYTES)					TAG 2 (27,000 BYTES)		TAG 3 (39,000 BYTES)							TAG 4 (42,000 BYTES)			

Note: Capacity of IXSHD is 99,999 bytes. Tags cannot be split; therefore, the first 3 tags fit into the IXSHD and the 4th tag is overflowed into the Data Extension Area.

D.3 Scope of implementation

BIIF describes the format of images, graphics, text and associated metadata within the BIIF file only. It does not define the image, graphics, or text requirements of the host system. The host system is responsible for the handling of unpacked image and text, as well as image, graphics and text display capabilities.

D.3.1 Creating headers and subheaders

This standard and associated profiles specify legal values for the header and subheader fields. The BIIF implementation for any particular host system will be responsible for enforcing the field values when creating files as stated in this standard.

D.3.2 Character counts

BIIF uses explicit byte counts to delimit fields. No end-of-field characters are used. These byte counts are critical for the proper interpretation of a BIIF file. The BIIF implementation should compute these byte counts based on the file contents to insure accuracy. All fields in BIIF header and subheaders must be present exactly as specified in BIIF header and subheader descriptions, and no additional fields may be inserted. BIIF uses various conditional fields whose presence, length and structure

is determined by previous fields and counts. If an expected conditional field is missing, the remainder of the file may be misinterpreted. A similar result will occur if a conditional field is inserted when it is not required. For these reasons, the item count fields are critical, and every effort must be made to ensure their accuracy.

D.3.3 Data entry

To reduce any operator workload imposed by the implementation, each implementation should provide where plausible for the automatic entry of data. Global default values for the particular BIIF version should be inserted automatically in the file. System default values, such as the standard size parameters for a base image, also should be entered automatically by the implementation. Values that are known to the system, such as the time or the computed size of an overlay, also should be entered automatically. Where operator value selection is needed, the use of pre-defined selection lists of valid values is encouraged. Range or value checking logic for operator input field entries will help avoid inadvertent operator entry errors.

D.3.4 Tagged Record Extensions

Users may need to add additional data to a BIIF file header or image subheader. To accommodate this requirement, user-defined data and extension fields are provided in the file header and segment subheader. One potential use for the user- defined data and extension subheader data is to provide space for directly associating acquisition parameters with the image. Use of these fields requires insertion of tagged records that implement the extension as described in this standard. Public tags shall be registered with the BIIF registration authority according to procedures available from the authority. This procedure ensures that different users will not use the same tag to flag different extended data. It also provides for configuration management of Tagged Record Extension formats where the extended data are expected to be used by a wide audience of users.

D.3.5 Out-of-bounds field values

The file creator is responsible for ensuring that all BIIF field values are within the bounds specified by BIIF document. An out-of-bounds value in a BIIF field indicates that either an error occurred or that the sending station was not in full compliance with BIIF.

D.3.6 Use of images

BIIF specifies a format for images contained within a BIIF file only. A BIIF implementation must be capable of translating this format to and from the host system's local format. Some host systems have multiple formats for binary data. In these cases, the BIIF implementation must use the appropriate host format to provide the necessary data exchange services with other system packages. When imagery data of less than M bits-per-pixel is displayed on an M-bit (2^M gray shades) display device, it must be transformed into the dynamic range of the device. One way to do this is to modify the LUTs of the display device. However, if M-bit and less than M-bit imagery is displayed simultaneously, the M-bit image will appear distorted. The recommended method is to convert the less than M-bit imagery into M-bit imagery, then use the standard LUTs. The following equation will transform a less than M-bit pixel into and M-bit pixel:

N = number of bits-per-pixel
P_N = N-bit pixel value
P_M = M-bit pixel value

$$P_M = \frac{2^M - 1}{2^N - 1} P_N$$

D.3.7 Use of text files

The text format field is provided to help the reader of the file determine how to interpret the text data received. The file reader is responsible for interpreting the various text formats. Format designations explicitly supported by BIIF are as follows:

D.3.7.1 BCS (TXTFMT=STA)

BIIF BCS is a specified format to provide a common format for all BIIF implementations. The BCS code shall be represented as depicted in Tables D-1 and D-2. This is the BCS code (Basic Latin Set) represented in ISO 646. The BCS codes shall be eight bits, a_1 through a_8. The eighth bit, a_8, shall be set to 0. A_8 shall be the Most Significant Bit (MSB), and a_1 shall be the Least Significant Bit (LSB). It is intended to provide for simple communications among BIIF stations. BIIF BCS-A format is comprised of the following BCS characters (all numbers are decimal): Line Feed (10), Form Feed (12), Carriage Return (13), and space (32) through tilde (126). This set includes all the alphanumeric characters as well as all commonly used punctuation characters. All lines within a BIIF STA file will be separated by carriage return/line feed pairs. It is the responsibility of the local system to translate these pairs into the local format. BIIF BCS has no standard line length. The host system must be capable of processing lines that are longer than the local standard.

D.3.7.2 Additional TXTFMT Codes

BIIF allows multi-octet codes to be contained in the text data field.. Different systems interpret these codes for various purposes. The code “STA” in the TXTFMT field shall mean that only BCS-A characters appear in the text data field. A TXTFMT value of UC2 indicates UCS-2, two octet UCS , UC4 is UCS-4, four octet UCS, UT1 is UTF-1 (Basic Latin and Latin-1 Supplement, Tables D1-D4), UCS Transformation Format 1, and UT8 is UTF-8, UCS Transformation Format 8 (Amendment 2 of ISO 10646).

Table D.2 -- Basic Latin character set

	000	001	002	003	004	005	006	007
0	000	016	SP 032	0 048	@ 064	P 080	` 096	p 112
1	001	017	! 033	1 049	A 065	Q 081	a 097	q 113
2	002	018	" 034	2 050	B 066	R 082	b 098	r 114
3	003	019	# 035	3 051	C 067	S 083	c 099	s 115
4	004	020	$ 036	4 052	D 068	T 084	d 100	t 116
5	005	021	% 037	5 053	E 069	U 085	e 101	u 117
6	006	022	& 038	6 054	F 070	V 086	f 102	v 118
7	007	023	' 039	7 055	G 071	W 087	g 103	w 119
8	008	024	(040	8 056	H 072	X 088	h 104	x 120
9	009	025	) 041	9 057	I 073	Y 089	i 105	y 121
A	010	026	* 042	: 058	J 074	Z 090	j 106	z 122
B	011	027	+ 043	; 059	K 075	[091	k 107	{ 123
C	012	028	, 044	< 060	L 076	\ 092	l 108	| 124
D	013	029	- 045	= 061	M 077	] 093	m 109	} 125
E	014	030	. 046	> 062	N 078	^ 094	n 110	~ 126
F	015	031	/ 047	? 063	O 079	_ 095	o 111	128

Table D.3 -- Basic latin character set explanation

Decimal	Hex	Name
032	20	SPACE
033	21	EXCLAMATION MARK
034	22	QUOTATION MARK
035	23	NUMBER SIGN
036	24	DOLLAR SIGN
037	25	PERCENT SIGN
038	26	AMPERSAND
039	27	APOSTROPHE
040	28	LEFT PARENTHESIS
041	29	RIGHT PARENTHESIS
042	2A	ASTERISK
043	2B	PLUS SIGN
044	2C	COMMA
045	2D	HYPHEN-MINUS
046	2E	FULL STOP
047	2F	SOLIQUS
048	30	DIGIT ZERO
049	31	DIGIT ONE
050	32	DIGIT TWO
051	33	DIGIT THREE
052	34	DIGIT FOUR
053	35	DIGIT FIVE
054	36	DIGIT SIX
055	37	DIGIT SEVEN
056	38	DIGIT EIGHT
057	39	DIGIT NINE
058	3A	COLON
059	3B	SEMICOLON
060	3C	LESS-THAN SIGN
061	3D	EQUALS SIGN
062	3E	GREATER-THAN SIGN
063	3F	QUESTION MARK
064	40	COMMERCIAL AT
065	41	LATIN CAPITAL LETTER A
066	42	LATIN CAPITAL B
067	43	LATIN CAPITAL C
068	44	LATIN CAPITAL D
069	45	LATIN CAPITAL E
070	46	LATIN CAPITAL F
071	47	LATIN CAPITAL G
072	48	LATIN CAPITAL H
073	49	LATIN CAPITAL I
074	4A	LATIN CAPITAL J
075	4B	LATIN CAPITAL K
076	4C	LATIN CAPITAL L
077	4D	LATIN CAPITAL M
078	4E	LATIN CAPITAL N
079	4F	LATIN CAPITAL O
080	50	LATIN CAPITAL P
081	51	LATIN CAPITAL Q

Table D.3 -- Basic Latin character set explanation
(concluded)

Decimal	Hex	Name
082	52	LATIN CAPITAL R
083	53	LATIN CAPITAL S
084	54	LATIN CAPITAL T
085	55	LATIN CAPITAL U
086	56	LATIN CAPITAL V
087	57	LATIN CAPITAL W
088	58	LATIN CAPITAL X
089	59	LATIN CAPITAL Y
090	5A	LATIN CAPITAL Z
091	5B	LEFT SQUARE BRACKET
092	5C	REVERSE SOLIDUS
093	5D	RIGHT SQUARE BRACKET
094	5E	CIRCUMFLEX ACCENT
095	5F	LOW LINE
096	60	GRAVE ACCENT
097	61	LATIN SMALL LETTER A
098	62	LATIN SMALL LETTER B
099	63	LATIN SMALL LETTER C
100	64	LATIN SMALL LETTER D
101	65	LATIN SMALL LETTER E
102	66	LATIN SMALL LETTER F
103	67	LATIN SMALL LETTER G
104	68	LATIN SMALL LETTER H
105	69	LATIN SMALL LETTER I
106	6A	LATIN SMALL LETTER J
107	6B	LATIN SMALL LETTER K
108	6C	LATIN SMALL LETTER L
109	6D	LATIN SMALL LETTER M
110	6E	LATIN SMALL LETTER N
111	6F	LATIN SMALL LETTER O
112	70	LATIN SMALL LETTER P
113	71	LATIN SMALL LETTER Q
114	72	LATIN SMALL LETTER R
115	73	LATIN SMALL LETTER S
116	74	LATIN SMALL LETTER T
117	75	LATIN SMALL LETTER U
118	76	LATIN SMALL LETTER V
119	77	LATIN SMALL LETTER W
120	78	LATIN SMALL LETTER X
121	79	LATIN SMALL LETTER Y
122	7A	LATIN SMALL LETTER Z
123	7B	LEFT CURLY BRACKET
124	7C	VERTICAL LINE
125	7D	RIGHT CURLY BRACKET
126	7E	TILDE

Table D.4 -- Latin-1 supplement character set

	008	009	00A	00B	00C	00D	00E	00F
0	128	144	NB SP 160	° 176	À 192	Ð 208	à 224	ð 240
1	129	145	¡ 161	± 177	Á 193	Ñ 209	á 225	ñ 241
2	130	146	¢ 162	² 178	Â 194	Ò 210	â 226	ò 242
3	131	147	£ 163	³ 179	Ã 195	Ó 211	ã 227	ó 243
4	132	148	¤ 164	´ 180	Ä 196	Ô 212	ä 228	ô 244
5	133	149	¥ 165	µ 181	Å 197	Õ 213	å 229	õ 245
6	134	150	¦ 166	¶ 182	Æ 198	Ö 214	æ 230	ö 246
7	135	151	§ 167	· 183	Ç 199	× 215	ç 231	÷ 247
8	136	152	¨ 168	¸ 184	È 200	Ø 216	è 232	ø 248
9	137	153	© 169	¹ 185	É 201	Ù 217	é 233	ù 249
A	138	154	ª 170	º 186	Ê 202	Ú 218	ê 234	ú 250
B	139	155	« 171	» 187	Ë 203	Û 219	ë 235	û 251
C	140	156	¬ 172	¼ 188	Ì 204	Ü 220	ì 236	ü 252
D	141	157	- 173	½ 189	Í 205	Ý 221	í 237	ý 253
E	142	158	® 174	¾ 190	Î 206	Þ 222	î 238	þ 254
F	143	159	¯ 175	¿ 191	Ï 207	ß_ 223	ï 239	ÿ 255

Table D.5 -- Latin-1 supplement character set explanation

Decimal	Hex	Name
160	A0	NO BREAK SPACE
161	A1	INVERTED EXCLAMATION MARK
162	A2	CENT SIGN
163	A3	POUND SIGN
164	A4	CURRENCY SIGN
165	A5	YEN SIGN
166	A6	BROKEN BAR
167	A7	SECTION SIGN
168	A8	DIAERESIS
169	A9	COPYRIGHT
170	AA	FEMININE ORDINAL INDICATOR
171	AB	LEFT-POINTING DOUBLE ANGLE QUOTATION MARK
172	AC	NOT SIGN
173	AD	SOFT HYPHEN
174	AE	REGISTERED SIGN
175	AF	MACRON
176	B0	DEGREE SIGN
177	B1	PLUS-MINUS SIGN
178	B2	SUPERSCRIPT TWO
179	B3	SUPERSCRIPT THREE
180	B4	ACUTE ACCENT
181	B5	MICRO SIGN
182	B6	PILCROW SIGN
183	B7	MIDDLE DOT
184	B8	CEDILLA
185	B9	SUPERSCRIPT ONE
186	BA	MASCULINE ORDINAL INDICATOR
187	BB	RIGHT POINTING DOUBLE ANGLE QUOTATION MARK
188	BC	VULGAR FRACTION ONE QUARTER
189	BD	VULGAR FRACTION ONE HALF
190	BE	VULGAR FRACTION THREE QUARTERS
191	BF	INVERTED QUESTION MARK
192	C0	LATIN CAPITAL LETTER A WITH GRAVE
193	C1	LATIN CAPITAL LETTER A WITH ACUTE
194	C2	LATIN CAPITAL LETTER A WITH CIRCUMFLEX
195	C3	LATIN CAPITAL LETTER A WITH TILDE
196	C4	LATIN CAPITAL LETTER A WITH DIAERESIS
197	C5	LATIN CAPITAL LETTER A WITH RING ABOVE
198	C6	LATIN CAPITAL LIGATURE AE
199	C7	LATIN CAPITAL LETTER C WITH CEDILLA
200	C8	LATIN CAPITAL LETTER E WITH GRAVE
201	C9	LATIN CAPITAL LETTER E WITH ACUTE
202	CA	LATIN CAPITAL LETTER E WITH CIRCUMFLEX
203	CB	LATIN CAPITAL LETTER E WITH DIAERESIS
204	CC	LATIN CAPITAL LETTER I WITH GRAVE
205	CD	LATIN CAPITAL LETTER I WITH ACUTE
206	CE	LATIN CAPITAL LETTER I WITH CIRCUMFLEX
207	CF	LATIN CAPITAL LETTER I WITH DIAERESIS
208	D0	LATIN CAPITAL LETTER ETH (ICELANDIC)
209	D1	LATIN CAPITAL N WITH TILDE
210	D2	LATIN CAPITAL LETTER O WITH GRAVE
211	D3	LATIN CAPITAL LETTER O WITH ACUTE
212	D4	LATIN CAPITAL LETTER O WITH CIRCUMFLEX

Table D.5 -- Latin-1 supplement character set explanation *(continued)*

Decimal	Hex	Name
213	D5	LATIN CAPITAL LETTER O WITH TILDE
214	D6	LATIN CAPITAL LETTER O WITH DIAERESIS
215	D7	MULTIPLICATION SIGN
216	D8	LATIN CAPITAL LETTER WITH STROKE
217	D9	LATIN CAPITAL LETTER U WITH GRAVE
218	DA	LATIN CAPITAL LETTER U WITH ACUTE
219	DB	LATIN CAPITAL LETTER U WITH CIRCUMFLEX
220	DC	LATIN CAPITAL LETTER U WITH DIAERESIS
221	DD	LATIN CAPITAL LETTER Y WITH ACUTE
222	DE	LATIN CAPITAL LETTER THORN (ICELANDIC)
223	DF	LATIN SMALL LETTER SHARP S (GERMAN)
224	E0	LATIN SMALL A WITH GRAVE
225	E1	LATIN SMALL LETTER A WITH ACUTE
226	E2	LATIN SMALL LETTER A WITH CIRCUMFLEX
227	E3	LATIN SMALL LETTER A WITH TILDE
228	E4	LATIN SMALL LETTER A WITH DIAERESIS
229	E5	LATIN SMALL LETTER A WITH RING ABOVE
230	E6	LATIN SMALL LIGATURE AE
231	E7	LATIN SMALL LETTER C WITH CEDILLA
232	E8	LATIN SMALL LETTER E WITH GRAVE
233	E9	LATIN SMALL LETTER E WITH ACUTE
234	EA	LATIN SMALL LETTER E WITH CIRCUMFLEX
235	EB	LATIN SMALL LETTER E WITH DIAERESIS
236	EC	LATIN SMALL LETTER I WITH GRAVE
237	ED	LATIN SMALL LETTER I WITH ACUTE
238	EE	LATIN SMALL LETTER I WITH CIRCUMFLEX
239	EF	LATIN SMALL LETTER I WITH DIAERESIS
240	F0	LATIN SMALL LETTER ETH (ICELANDIC)
241	F1	LATIN SMALL LETTER N WITH TILDE
242	F2	LATIN SMALL LETTER O WITH GRAVE
243	F3	LATIN SMALL LETTER O WITH ACUTE
244	F4	LATIN SMALL LETTER O WITH CIRCUMFLEX
245	F5	LATIN SMALL LETTER O WITH TILDE
246	F6	LATIN SMALL LETTER O WITH DIAERESIS
247	F7	DIVISION SIGN
248	F8	LATIN SMALL LETTER O WITH STROKE
249	F9	LATIN SMALL LETTER U WITH GRAVE
250	FA	LATIN SMALL LETTER U WITH ACUTE
251	FB	LATIN SMALL LETTER U WITH CIRCUMFLEX
252	FC	LATIN SMALL LETTER U WITH DIAERESIS
253	FD	LATIN SMALL LETTER Y WITH ACUTE
254	FE	LATIN SMALL LETTER THORN (ICELANDIC)
255	FF	LATIN SMALL LETTER Y WITH DIAERESIS

D.3.8 File system constraints

A BIIF file is presented as a stream of contiguous octets. The file length value which appears in the file header represents an exact count of the meaningful octets contained within the BIIF file structure. This value may not always correspond with file size values reported by some computer file storage systems (e.g. those that store files on multi-byte block boundaries vice single byte boundaries). The requirement for storing and exchanging BIIF files between systems with differing file storage structures must be accommodated when implementing BIIF. When the file storage system reports a larger file size than the value contained in the file header, the file length value in the file is the basis for determining the last byte of meaningful data within the file. When a smaller file size is reported by the file system, it is an indication that the file has been inadvertently truncated during the file exchange process and is therefore incomplete or otherwise defective.

D.3.9 Security considerations

A BIIF file contains sufficient security information in the file header, image and graphic subheaders to allow implementors to meet virtually any security requirement for controlling the use and presentation of data. Exact security information handling requirements generally are specified by appropriate accreditation authorities or specific user requirements. It is suggested that implementors extract the appropriate security markings from one or more of the header/subheaders and ensure that the information is always displayed whenever the pertinent part of BIIF file is displayed.

D.4 Product configurations

BIIF provides a very flexible means to package imagery products. Interoperability places a significant burden on BIIF read capable implementations to be able to interpret and use potentially any combination and permutation of BIIF file format options that may be created by BIIF file producers. Consequently, significant care should be taken when defining product specifications for BIIF formatted imagery products.

The objective of the following discussion is to describe several generalized product configurations that can be used as the basis for defining specific imagery products. These product configurations are typical of those successfully used within the imagery and mapping community to date.

D.4.1 General

An imagery product may potentially be produced under one of the following concepts:

1. Single File, Single Base Image - This is the most common use of BIIF format. In this product concept, the BIIF file is produced with a focus on a single image, commonly called the ‘base image’. All other segments and extended data within the file are focused on amplifying the information portrayed in the base image.
2. Single File, Multiple Images - In this product concept, the BIIF file is produced containing multiple images, all of which have equal or similar significance to the value of the product. Other segments and extended data within the file are focused on amplifying the information portrayed in the image(s) to which they are associated.
3. Single File, No Image - This type of product may only have graphic segments, or only text segments, or only extension segments, or any combination of these segments. The significance of the data within the file may pertain only to that file, or it may pertain to one or more files with which it is associated.
4. Multiple Correlated Files - For this product concept, the product is comprised of multiple BIIF files that are interrelated as explicitly defined in the product specification.

D.4.1.1 Single file, single base image

For this type of product file, there is one image of central focus, the base image, placed on the Common Coordinate System (CCS) plane. Its first pixel may be located at the origin (0,0) of the CCS or off-set from the CCS origin according to the row/column coordinate values placed in the location (LOC) field of the image subheader. Figure D.1 provides a representative portrayal for the following discussion.

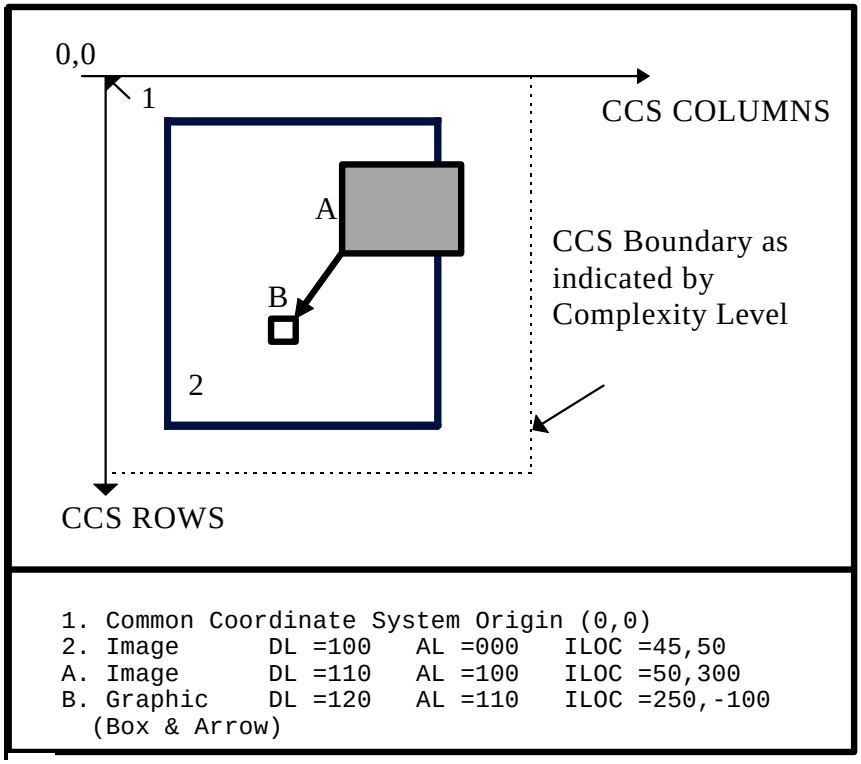


Figure D.1 -- Single file, single base image representation

D.4.1.1.1 Image segment overlays

Additional images, often called subimages or inset images, may be included as separate image segments in the file. The purpose of these images is to add information or clarity about the base image. Their placement in the CCS plane is controlled by the value of each segment's Attachment Level (AL) and Location (LOC) row/column value. When overlay images are attached to the base image, the LOC value represents a row/column off-set in the CCS from the location specified by the base image row/column LOC value. If the overlay image is unattached to any other segment (AL=000), the overlay's LOC value is a row/column off-set from the CCS origin (0,0).

D.4.1.1.2 Graphic segment overlays

Graphic Segments are used to provide graphical (lines, polygons, ellipses, etc.) and textual annotation to the base image. The graphic representation is done using Computer Graphics Metafile (CGM). In a manner similar to image segment overlays, the placement of graphics in the CCS plane is controlled by the value of each segment's AL and LOC values. CGM has its own internal coordinate space called "Virtual Display Coordinates (VDC)" that has its own defined origin (0,0) point. The row/column value in the graphic segment LOC field identifies the placement of the graphic's VDC origin point relative to the CCS origin when AL=000 or relative to the segment LOC to which it is attached.

D.4.1.1.3 Non-destructive overlays

BIIF image and graphic segment overlays are handled in a non-destructive manner. The overlays may be placed anywhere within the bounds of the CCS (defined for a specific BIIF file by the Complexity level (Clevel)). They may be placed totally on the base image, partially on the base image, or entirely off of the base image. Any image or graphic segment can be placed on or under any other segment, fully or partially. The visibility of pixel values of overlapping segments is determined by the Display Level (DL) assigned to that segment. Each displayable segment (images and graphics) is assigned a DL (ranging from 001 - 999) that is unique within the file. At any CCS pixel location shared by more than one image or graphic, the visible pixel value is the one from the segment having the greatest DL value. If the user of an BIIF file opts to move an overlay or turn off the presentation of an overlay, the next greatest underlying pixel value(s) will then become visible. This approach allows for the non-

destructible nature of BIIF overlays as opposed to the 'burned in' approach where overlay pixel values are used to replace pixels values of the underlying image.

D.4.1.1.4 Text Segments

Text segments allow inclusion in BIIF file of textual information related to the base image, perhaps a textual description of the activities portrayed in the image.

D.4.1.1.5 Extension data

BIIF file header and each standard data type sub-header have designated expandable fields to allow for the optional inclusion of extension data. The inclusion of extension data provides the ability to add data/information about the standard data type (metadata) that is not contained in the basic fields of the headers and subheaders. The additional data is contained within one or more BIIF tagged record extensions that are placed in the appropriate field (user defined data field or extended data field) of the standard data type subheader for which the metadata applies. When tagged record extensions have application across multiple data types in the file, or otherwise apply to the entire BIIF file in general, they are placed in the appropriate file header fields.

Whereas general purpose BIIF readers should always be able to portray image and graphic segments and act on standard header and subheader data, they may not always be able to act on product specific extension data. Upon receipt of a file that contains extension data, a BIIF compliant system should at least ignore the extensions and properly interpret the other legal components of BIIF file. Exemplary use of tagged record extensions:

1. Data about people, buildings, places, landmarks, equipment or other objects that may appear in the image
2. Data to allow correlation of information among multiple images and annotations within a BIIF file
3. Data about the equipment settings used to obtain the digital image, xray, etc.
4. Data to allow geopositioning of items in the imagery or measurement of distances of items in the imagery.

D.4.1.2 Single file, multiple images

For this type of product file, multiple images of equal or similar focus (multiple 'base' images) are placed within the Common Coordinate System (CCS) plane. Each image is located at an off-set from the CCS origin such that there is no overlap among the images. The Complexity Level of the file must be chosen such that the bounds of the CCS for the file are sufficient to contain the extent of all segments within the file. Figure D.2 provides a representative portrayal for this product type.

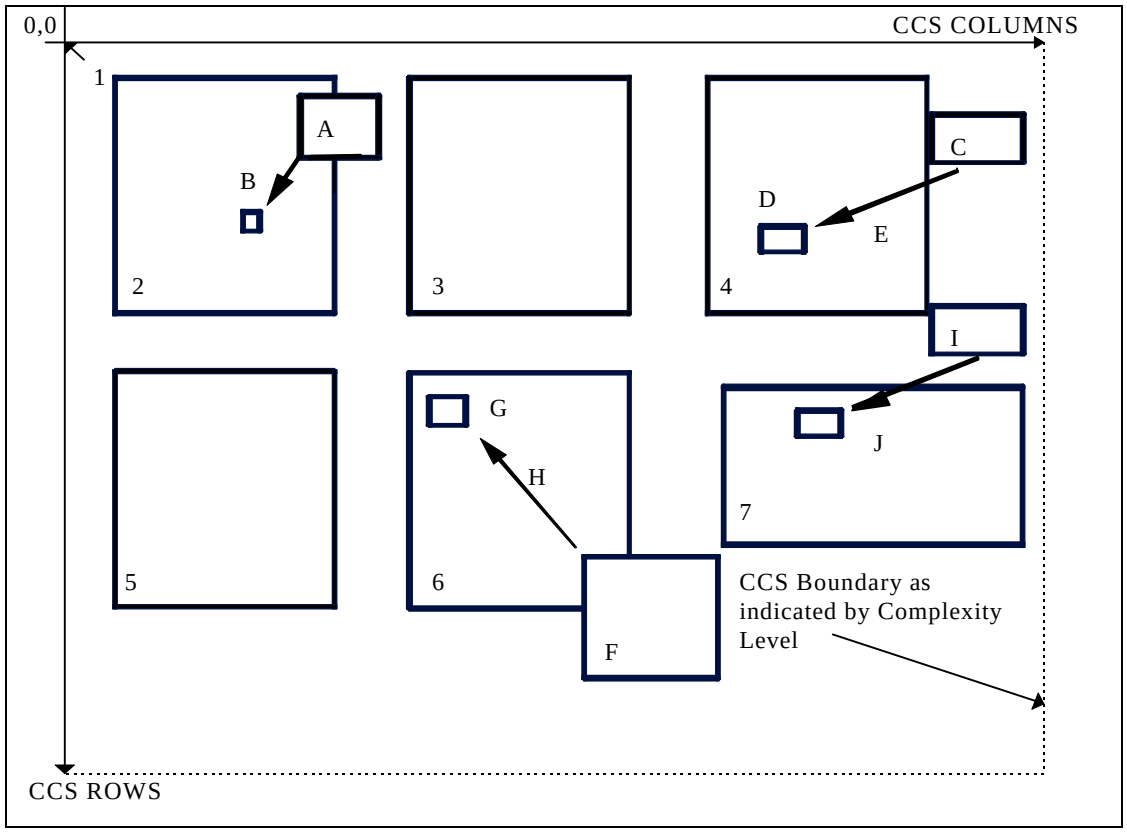


Figure D.2 -- Single file, multiple images representation

	Display Level	Attachment Level	Location
1. Common Coordinate System Origin (0,0)			
2. Image	DL = 100	AL = 000	50, 50
(A) Image Segment Overlay	DL = 110	AL = 100	50, 300
(B) Graphic Overlay (CGM Rectangle & Arrow)	DL = 120	AL = 110	50, 225
3. Image	DL = 200	AL = 000	50, 550
4. Image	DL = 300	AL = 000	50,1050
(C) Image Segment Overlay	DL = 310	AL = 300	100, 450
(D) Graphic Segment Overlay (CGM Rectangle)	DL = 320	AL = 300	300, 75
(E) Graphic Segment Overlay (CGM Arrow)	DL = 330	AL = 300	150, 100
5 Image	DL = 400	AL = 000	800, 50
6 Image	DL = 500	AL = 000	800, 550
(F) Image Segment Overlay	DL = 510	AL = 500	300, 250
(G) Graphic Segment Overlay (CGM Rectangle)	DL = 520	AL = 500	35, 35
(H) Graphic Segment Overlay (CGM Arrow)	DL = 530	AL = 500	125, 125
7 Image	DL = 600	AL = 000	820,1050
(I) Image Segment Overlay	DL = 610	AL = 600	-150, 400
(J) CGM Graphic Overlay (Rectangle & Arrow)	DL = 620	AL = 600	50, 100

Figure D.2 -- Single file, multiple images representation *(concluded)*

D.4.1.2.1 Overlays

Each image may be overlaid with additional image and graphic segments in the same fashion as described for the single file, single image case above. All overlays associated with a specific image should be attached to that specific image. Display Levels assigned to each image and graphic in the file must be unique within the file.

D.4.1.2.2 Text segments

Each text segment should be clearly marked as to whether it applies to the file as a whole or if it is associated with specific images within the file.

D.4.1.2.3 Extension data

Tagged record extensions are placed in the file header extension fields when applicable to the file as a whole. Extensions specific to a segment are placed in that segment's subheader.

D.4.1.3 Single file, no image

A BIIF single file product does not always contain an image. The BIIF file could contain zero or more graphic segments, zero or more text segments, zero or more extension segments, or any combination of at least one of these, without containing any image segments in the file. The file may be useful as a stand alone product, or it may be intended for use in conjunction with other BIIF files. For example, the file could contain graphic overlays to be merged with or applied to another BIIF file that was pre-positioned or transmitted at an earlier time. Any general purpose BIIF reader should at least be able to interpret and render the standard segments of a BIIF file that does not contain any image segments on a stand alone basis.

D.4.1.4 Multiple correlated files

An imagery product may be comprised of multiple BIIF files that are interrelated in a specified manner. This approach vastly increases the potential combination and permutation of options a general purpose BIIF reader would need to support to maintain full interpret capability. Therefore, each BIIF file in a multiple correlated file set must be structured such that a general purpose BIIF reader can properly interpret and render the file as if it were a stand alone product. The correlation of multiple BIIF files in a single product must be explicitly and unambiguously defined in a product specification. BIIF readers can then be further categorized according to specific multiple file product specifications that are supported. Representative use of multiple correlated BIIF files includes:

D.4.1.4.1 Stereo imagery

Some stereo image products are comprised of separate BIIF files for the stereo components of each image scene.

D.4.1.4.2 Imagery mosaics

Some extremely large image and map products consist of multiple BIIF files structured such that they can be pieced together in mosaic fashion by the interpret application as if the multiple files were a single larger image.

D.4.1.4.3 Reduced resolution data sets (rsets)

Rset products are comprised of multiple BIIF files. One file contains a full resolution image, and the other files contain the same image in a variety of lower resolutions.

D.4.1.4.4 Imagery and maps

Geopositioning products exist which consist of multiple separate BIIF files containing interrelated maps, images, graphics, legends, product indices, and georeference data.

ANNEX E (informative)
Examples BIIF Profiles

E.1 Example file.

E.1.1 Use of this example

Though the example was conceived initially to support the transmission of a file composed of a single base image, image insets (subimage overlays), graphic overlays, and text, its current form makes it suitable for a wide variety of file exchange needs. One of the flexible features of the example is that it allows several items of each data type to be included in one file, yet any of the data types may be omitted. Thus, for example, the example may equally well be used for the storage of a single portion of text, a single image or a complex composition of several images, graphics, and text. The following subclause discusses an example file of moderate complexity.

E.1.2 BIIF Model Profile

Table I shows the contents of the fields in the header of an example file composed of two image segments (one base image, one inset image), five graphic overlays, and five text selections. Figure E.1 shows part of the sample file as a composite image with its overlay graphics. In this example file, the data for each data item is preceded by the item's subheader. The subheader for a data type is omitted if no items of that type are included in the file. Subheader field contents for items in the sample file are shown in Tables E.1 through E.6.

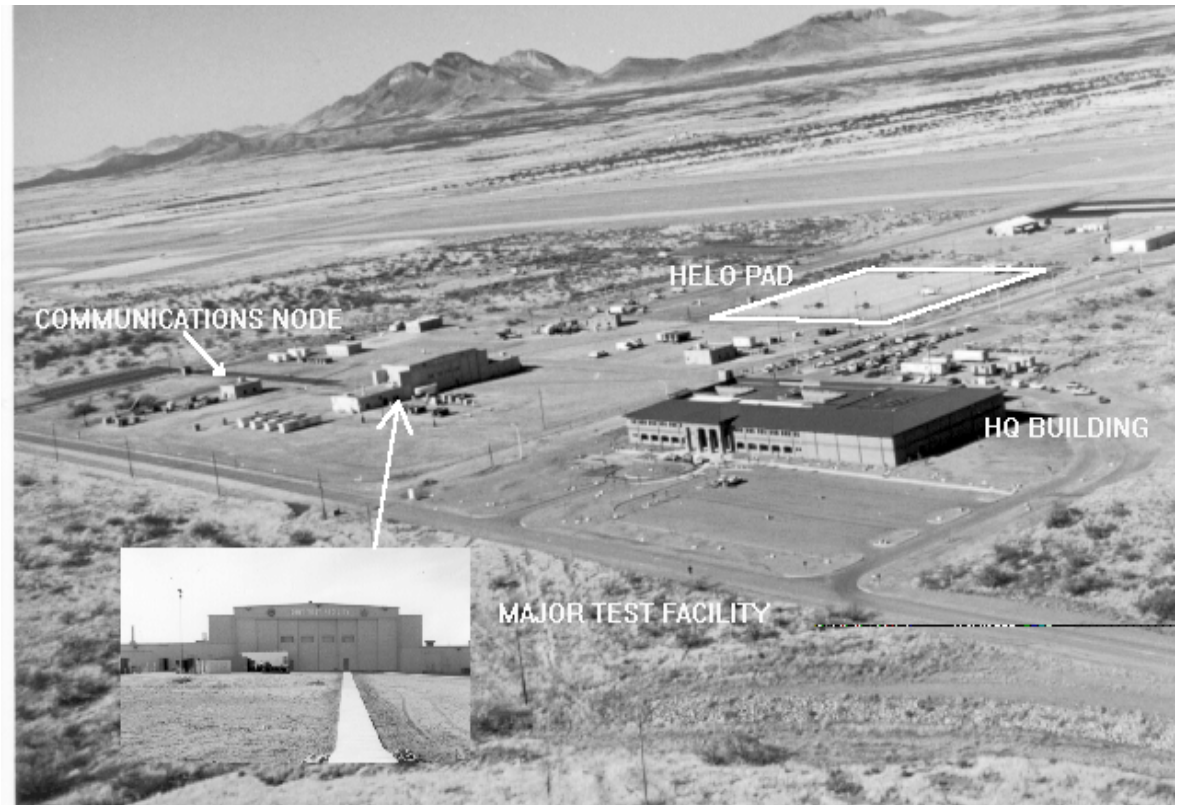


Figure E.1 -- Sample file composite image

Table E.1 -- Example BIIF file header

HEADER FIELD	FORMAT	COMMENT
Profile Name & Version (FHDR)	BFMP01.00	9 characters
Complexity Level (CLEVEL)	02	2 characters --extent of common coordinate system 2k x 2k
Standard Type (STYPE)	BF01	4 characters
Originating System ID (OSTAID)	U21SOO90	8 characters with 2 spaces
File Date & Time (FDT)	19960930224632	14 characters
File Title (FTITLE)	MAJOR TEST FACILITY	19 characters followed by 61 spaces - 80 characters
File Security Parameters (SEC1)	U	1 character
File Security Parameters (SEC2)		166 spaces
FSCOP	00000	5 zeros
FSCPYS	00000	5 zeros
Encryption (ENCRYP)	0	no encryption
Originator's ID (OID)	John Brown 44 1480 84 5611	26 characters followed by 19 spaces - 45 characters
File Length (FL)	000002925155	12 digits
Header Length (HL)	000515	6 digits
Number of Images (NUMI)	002	3 digits
Length of 1st Image Subheader (LISH001)	000679	6 digits
Length of 1st Image (LI001)	0002730600	10 digits
Length of 2nd Image Subheader (LISH002)	000439	6 digits
Length of 2nd Image (LI002)	0000089600	10 digits
Number of Symbols (NUMS)	005	3 digits
Length of 1st Symbol Subheader (LSSH001)	0258	4 digits
Length of 1st Symbol (LS001)	000122	6 digits
Length of 2nd Symbol Subheader (LSSH002)	0258	4 digits
Length of 2nd Symbol (LS002)	000122	6 digits
Length of 3rd Symbol Subheader (LSSH003)	0258	4 digits
Length of 3rd Symbol (LS003)	000150	6 digits
Length of 4th Symbol Subheader (LSSH004)	0258	4 digits
Length of 4th Symbol (LS004)	000112	6 digits

Table E.1 -- Example BIIF file header *(concluded)*

HEADER FIELD	FORMAT	COMMENT
Length of 5th Symbol Subheader (LSSH005)	0258	4 digits
Length of 5th Symbol (LS005)	000116	6 digits
Reserved for future use (NUMX)	000	3 digits
Number of Text Item (NUMT)	005	3 digits
Length of 1st Text Subheader (LTSH001)	0282	4 digits
Length of 1st Text Item (LT001)	20000	5 digits
Length of 2nd Text Subheader (LTSH002)	0282	4 digits
Length of 2nd Text Item (LT002)	20000	5 digits
Length of 3rd Text Subheader (LTSH003)	0282	4 digits
Length of 3rd Text Item (LT003)	20000	5 digits
Length of 4th Text Subheader (LTSH004)	0282	4 digits
Length of 4th Text Item (LT004)	20000	5 digits
Length of 5th Text Subheader (LTSH005)	0282	4 digits
Length of 5th Text Item (LT005)	20000	5 digits
Number of Data Extension Segments (NUMDES)	000	3 digits
Number of Reserved Data Extension Segments (NUMRES)	000	3 digits
User Defined Header Data Length (UDHDL)	00000	5 digits
Extended Header Data Header Length (XHDL)	00000	5 digits

E.1.2.1 Explanation of the file header

The Profile name and Version, BFMP01.00, is listed first. The next field contains the file's Complexity Level, in this case 02. A four character field for the Standard Type, BF01, appears next. An identification code containing ten characters for the System originating the primary information in the file is given next. The file origination date and time follow this and are given in UTC (Zulu) time format. This is followed by the File Title field containing up to 80 characters of free form text. The title of the sample file contains less than 80 characters, and, therefore, the remainder of the field is filled with blanks.

The File Security designator for “unspecified” follows and contains one character. Security-parameter optional fields (blank) and field (no copy tracking) follow. File Encryption is given a "0" indicating that the file is not encrypted. The originator's name and phone number are given next. These fields may be left blank. Then the length in bytes of the entire file is given, including all headers, subheaders, and data. This is followed by the length in bytes of the BIIF file header.

The Number of Images field contains the characters 002 to indicate two images are included in the file. This is followed by six characters to specify the length of the first image subheader, then ten characters for the length of the first image. The length of

the second image subheader and the length of the second image follow. The next item in the file header is the Number of Symbols, which contains 005 to indicate that five symbols are present in the file. The next fifty characters contain the Length of Symbol Subheader and Length of Symbol (four and six characters respectively) for the first through fifth symbol, one after the other.

The field, Number of Text Segments, is given as 005 and is followed by four characters specifying the length of the text subheader and five characters specifying the number of characters in the text segment for each of the five text segments. The Number of Data Extension Segments and Number of Reserved Extension Segments fields are given as "000." This completes the "roadmap" for separating the data subheaders from the actual data to follow.

The next fields in the header is the User Defined Header Data Length. User defined data could be used to include tagged record extensions that provide additional information about the file. In this example, however, the length of the user-defined header data is given as zero; therefore, the User Defined Header Data Field is omitted. The last field in the header is the Extended Header Data Length. The length of the extended header is given as zero; therefore, the Extended Header Data field is omitted, indicating that no tagged record extensions are included in this part of file header.

E.1.2.2 Explanation of the image subheaders

E.1.2.2.1 Explanation of the first image subheader

There are two images in this sample file. The first image has Display Level 001. Its subheader is shown in Table E.2. It is an unclassified, single band, single block, grey scale image with 8 bits per pixel and does not have an associated LUT. There are three associated comments. It is visible imagery, does not have geo-location data, and is stored as an uncompressed image. It is located at the origin of the common coordinate system within which all the displayable file components are located. It is 1332 rows by 2050 columns. Figure 15 illustrates the image printed at approximately three hundred pixels per inch.

Table E.2 -- Example of the first image subheader

IMAGE SUBHEADER FIELD	FORMAT	COMMENT
File Part Type (IM)	IM	2 characters
Image ID (IID)	0000000001	10 characters
Image Date & Time (IDATIM)	19960825203147	14 characters
Item Information (IINFO)	RE12345678AAAAAUS Major Test Facility and HQ	44 characters followed by 53 spaces - 97 characters total
Image Security Parameter (SEC1)	U	1 character
Image Security Parameters (SEC2)		166 spaces
Encryption (ENCRYP)	0	Not encrypted
Image Source (ISORCE)	Hand-held digital camera model XYZ.	35 characters followed by 7 spaces - 42 total characters
Number of Significant Rows in image (NROWS)	00001332	8 characters
Number of Significant Columns in image (NCOLS)	00002050	8 characters
Pixel Value Type (PVTYPE)	INT	3 characters - interpret pixel values as integers
Image Representation (IREP)	MONO	4 characters followed by 4 spaces - greyscale imagery
Image Class (ICAT)	VIS	3 characters followed by 5 spaces - visible imagery
Actual Bits-Per-Pixel Per Band (ABPP)	08	2 digits
Pixel Justification (PJUST)	R	1 character
Image Coordinate Type (ICORDS)		1 Space - indicates no geo location coordinates
Number of Image Comments (NICOM)	3	1 digit

Table E.2 -- Example of the first image subheader *(continued)*

IMAGE SUBHEADER FIELD	FORMAT	COMMENT
* Image Comment 1 (ICOM1)	This is a comment on Major Test Facility base and associated inset.	67 character plus 12 spaces 80 total characters
* Image Comment 2 (ICOM2)	It was developed at Fort Huachuca, Arizona. It shows the Joint Interoperability T.	80 total characters
* Image Comment 3 (ICOM3)	est Command Building and associated range areas.	49 characters followed by 31 spaces - 80 total characters
Image Compression (IC)	NC	2 characters - indicates no compression
Number of Bands (NBANDS)	1	1 digit
1st Band Representation (IREPBAND1)		2 spaces
1st Band Significance for Image Category (ISUBCAT1)		6 spaces
1st Band Image Filter Condition (IFC1)	N	1 character - N = no filter condition value
1st Band Standard Image Filter Code (IMFLT1)		3 spaces
1st Band Number of LUTS (NLUTS1)	0	1 character - No LUTS
Image Sync Code (ISYNC)	0	1 digit - No Sync Code
Image Mode (IMODE)	B	1 character - B required for 1 band
Number of Blocks per Row (NBPR)	0001	4 digits
Number of Blocks per Column (NBPC)	0001	4 digits
Number of pixels Per Block Horizontal (NPPBH)	2050	4 digits
Number of pixels Per Block Vertical (NPPBV)	1332	4 digits
Number of Bits per Pixel (NBPP)	08	2 digits
Display Level (IDLVL)	001	3 characters - display level 1
Attachment Level (IALVL)	000	Required 3 digit value since minimum display level

Table E.2 -- Example of the first image subheader *(concluded)*

IMAGE SUBHEADER FIELD	FORMAT	COMMENT
Location (ILOC)	0000000000	10 characters upper left pixel located at origin of common coordinate system
Image magnification (IMAG)	1.0	3 character followed by a space - 4 characters total
User Defined Image Data Length (UDIDL)	00000	5 digits
Extended Subheader Data Length (IXSHDL)	00000	5 digits

* According to the standard - this can look like a single contiguous comment of up to 3 x 80 characters or can be treated as separate image comments, 80 characters each.

E.1.2.2.2 Explanation of the second image subheader

The second image subheader in the file is depicted in Table E.3. As is the first image, this image is an 8-bit visible, grey scale image. It is much smaller (400 columns x 224 rows) and is not compressed. Also, unlike the first image, it has no associated comment fields, indicated by the fact NICOM = 0. Since it is attached to the first image (IALVL = 001), the ILOC field reveals that this image is located with its upper left corner positioned at Row 578, Column 142 with respect to the upper left corner of the first image. Since it has a display level greater than that of the first image, it will obscure part of the first image when they are both displayed.

Table E.3 -- Example of the second image subheader

IMAGE SUBHEADER FIELD	FORMAT	COMMENT
File Part Type (IM)	IM	2 characters
Image ID (IID)	0000000002	10 characters
Image Date & Time (IDATIM)	19960927011729	14 characters
ITEM INFORMATION (IIINFO)	BE12345678AAAAAA US Test Facility	29 characters followed by 68 spaces 97 characters total
Image Title (ITITLE)	Zoomed Test Facility	20 characters followed by 60 spaces - 80 characters total
Image Security Parameter (SEC1))	U	1 character
Image Security Parameters (SEC2)		166 spaces
Encryption (ENCRYP)	0	Not encrypted
Image Source (ISORCE)	Cut of original image	21 characters followed by 21 spaces - 42 characters total
Number of Significant Rows in image (NROWS)	00000224	8 characters
Number of Significant Columns in image (NCOLS)	00000400	8 characters
Pixel value type (PVTYPE)	INT	3 characters - interpret pixel values as integers
Image Representation (IREP)	MONO	4 characters followed by 4 spaces - grey scale imagery

Table E.3 -- Example of the second image subheader *(concluded)*

IMAGE SUBHEADER FIELD	FORMAT	COMMENT
Image Class (ICAT)	VIS	3 characters followed by 5 spaces - visible imagery
Actual Bits-Per-Pixel Per Band (ABPP)	08	2 digits
Pixel Justification (PJUST)	R	1 character
Image Coordinate System (ICORDS)		1 Space - indicates no geo location coordinates
Number of Image Comments (NICOM)	0	1 digit no comments
Image Compression (IC)	NC	2 chars - indicates uncompressed
Number of Bands (NBANDS)	1	1 digit
1st Band Representation (IREPBAND1)		2 spaces
1st Band Significance (ISUBCAT1)		6 spaces
1st Band Image Filter Condition (IFC1)	N	1 character - no filter condition
1st Band Standard Image Filter Code (IMFLT1)		3 spaces
1st Band Number of LUTS (NLUTS1)	0	1 character - no LUTS
Image Sync Code (ISYNC)	0	1 digit - no Sync Code
Image Mode (IMODE)	B	1 character - B required for 1 band
Number of Blocks per Row (NBPR)	0001	4 digits
Number of Blocks per Column (NBPC)	0001	4 digits
Number of pixels Per Block Horizontal (NPPBH)	0400	4 digits
Number of pixels Per Block Vertical (NPPBV)	0224	4 digits
Number Bits Per Pixel (NBPP)	08	2 digits
Display Level (IDLVL)	002	3 digits
Attachment Level (IALVL)	001	3 digits
Location (ILOC)	0057800142	10 characters, located at row 578 column 142 of first image
Image Magnification (IMAG)	1.0	3 characters followed by a space - 4 characters total
User Defined Image Data Length (UDIDL)	00000	5 digits
Extended Subheader Data Length (IXSHDL)	00000	5 digits

E.1.2.3 Explanation of the symbol subheaders

E.1.2.3.1 Explanation of the first symbol subheader

This symbol is a computer graphics metafile graphic (HELO PAD RECTANGLE). The symbol is attached to the first image, and its location is recorded in SLOC (row 392, column 1110) and is measured as an offset from the origin at the upper left corner of that image.

Table E.4 -- Symbol subheader for the first symbol

SYMBOL SUBHEADER FIELD	FORMAT	COMMENT
File Part Type (SY)	SY	2
Symbol ID (SID)	0000000001	10
Symbol Name (SNAME)	HELO PAD RECTANGLE	18 characters followed by 2 spaces - total 20 characters
Symbol Security Parameter (SEC1)	U	1 character
Symbol Parameter (SEC2)		166 spaces
Encryption (ENCRYP)	0	Not encrypted
Symbol Format (SFMT))	C	1 character - indicates CGM
Symbol Structure (STRUCT)		13 spaces
Display Level (SDLVL)	003	3 digits
Attachment Level (SALVL)	001	3 digits
Symbol Location (SLOC)	0039201110	10 characters
Symbol Location (SSLOC2)	0000000000	10 characters (zeros)
Symbol Parameters (SCOLOR)	M	indicates CGM file contains no color components
Symbol Parameters (SRES2)	000000000000	12 characters (zeros)
Extended Subheader Data Length (SXSHDL)	00000	5 digits

E.1.2.3.2 Explanation of the second symbol subheader

The second symbol is also a CGM graphic. It is the arrow pointing to the test facility. It is attached to the subimage. Therefore, its location as recorded in SLOC is measured as an offset from the upper left corner of the subimage.

Table E.5 -- Symbol subheader for the second symbol

SYMBOL SUBHEADER FIELD	FORMAT	COMMENT
File Part Type (SY)	SY	2
Symbol ID (SID)	0000000002	10
Symbol Name (SNAME)	ARROW	5 characters followed by 15 spaces - total 20 characters
Symbol Security Parameter (SEC1)	U	1 character
Symbol Security Parameters (SEC2))		166 spaces
Encryption (ENCRYP)	0	No Encryption
Symbol Format (SFMT)	C	1 character - indicates CGM
Symbol Structure (SSTRUCT)		13 spaces
Display Level (SDLVL)	004	3 digits
Attachment Level (SALVL)	002	3 digits
Symbol Location (SLOC)	0000000285	10 chars relative to origin of 2nd image
Symbol Location (SLOC2)	000000000000	12 characters (zeros)
Symbol Parameters (SCOLOR)	M	indicates CGM file contains no color components
Symbol Parameters (SRES2)	000000000000	12 characters (zeros)
Extended Subheader Data Length (SXSHDL)	00000	5 digits

E.1.2.3.3 Explanation of the third Symbol subheader

The third Symbol is a CGM text annotation (HQ Building). It is attached to the base image. Its location as recorded in SLOC is measured as an offset from the upper left corner of the base image, in this case SLOC is 0,0 and the offsetting for this Symbol is actually done within the CGM construct itself.

Table E.6 -- Symbol subheader for the third symbol

SYMBOL SUBHEADER FIELD	FORMAT	COMMENT
File Part Type (SY)	SY	2
Symbol ID (SID)	0000000003	10
Symbol Name (SNAME)	HQ BUILDING	11 characters followed by 9 spaces - total 20 characters
Symbol Security Parameter (SEC1)	U	1 character
Symbol Security parameters (SEC2)		166 spaces
Encryption (ENCRYP)	0	No Encryption
Symbol Format (SFMT))	C	1 character - indicates CGM
Symbol Structure (SSTRUCT)		13 spaces
Display Level (SDLVL)	005	3 digits
Attachment Level (SALVL)	001	3 digits
Symbol Location (SLOC)	0000000000	10 characters
Symbol Location (SLOC2)	0000000000	10 characters (zeros)
Symbol Parameter (SCOLOR)	M	indicates CGM file contains no color components
Symbol Parameter(SRES2)	000000000000	12 characters (zeros)
Extended Subheader Data Length (SXSHDL)	00000	5 digits

E.1.2.3.4 Explanation of the fourth symbol subheader

The fourth symbol is a CGM annotation text. It is the MAJOR TEST FACILITY text. It is attached to the subimage. Therefore, its location as recorded in SLOC is placed as an offset from the upper left corner of the subimage.

Table E.7 -- Symbol subheader for the fourth symbol.

SYMBOL SUBHEADER FIELD	FORMAT	COMMENT
File Part Type (SY)	SY	2
Symbol ID (SID)	0000000004	10
Symbol Name (SNAME)	MAJOR TEST FACILITY	19 characters followed by 1 spaces - total 20 characters
Symbol Security Parameter (SEC1)	U	1 character
Symbol Security Parameters (SEC2))		166 spaces
Encryption (ENCRYP)	0	No Encryption
Symbol Format (SFMT)	C	1 character - indicates CGM
Symbol Structure (SSTRUCT)		13 spaces
Display Level (SDLVL)	006	3 digits
Attachment Level (SALVL)	002	3 digits
Symbol Location (SLOC)	0008500415	10 characters
Symbol Location (SLOC2)	0000000000	10 characters
Symbol Parameter (SCOLOR)	M	indicates CGM file contains no color components
Symbol Parameters (SRES2)	000000000000	12 spaces
Extended Subheader Data Length (SXSHDL)	00000	5 digits

E.1.2.3.5 Explanation of the fifth symbol subheader

The fifth symbol is a CGM symbol with text annotation. It is the COMMUNICATIONS NODE annotation with associated arrow. It is attached to the base image. Therefore, its location as recorded in SLOC is placed as an offset from the upper left corner of the base image.

Table E.8 -- Symbol subheader for the fifth symbol

SYMBOL SUBHEADER FIELD	FORMAT	COMMENT
File Part Type (SY)	SY	2
Symbol ID (SID)	0000000005	10
Symbol Name (SNAME)	COMMUNICATION ARROW	19 characters followed by 1 spaces - total 20 characters
Symbol Security Parameter (SEC1)	U	1 character
Symbol Security Parameters (SEC2))		166 spaces
Encryption (ENCRYP)	0	No Encryption
Symbol Format (SFMT)	C	1 character - indicates CGM
Symbol Structure (SSTRUCT)		13 spaces
Display Level (SDLVL)	007	3 digits
Attachment Level (SALVL)	001	3 digits
Symbol Location (SLOC)	0047000040	10 characters
Symbol Location (SLOC2)	0000000000	10 characters
Symbol Parameter (SCOLOR)	M	indicates CGM file contains no color components
Symbol parameter(SRES2)	000000000000	12 characters (zeros)
Extended Subheader Data Length (SXSHDL)	00000	5 digits

E.1.2.4 Explanation of the text subheaders

There are 5 text items included in the file. Other than the text data they contain, they differ only in matters such as title, date-time of creation, and ID. Therefore, only the first is discussed since the subheaders of all the rest are essentially the same.

E.1.2.4.1 Explanation of the first text subheader

The first text document was created on September 30, 1996 at 22:45 hours. Its subheader is shown in Table E.9.

Table E.9 -- Text subheader for the text document

TEXT SUBHEADER FIELD	FORMAT	COMMENT
File Part Type (TE)	TE	2 characters
Text ID (TEXTID)	0000000001	10 characters
Text Date & Time (TXTDT)	19960930224530	14 characters
Text Title (TXTITL)	First sample text file.	23 characters followed by 57 spaces - 80 total characters
Text Security Parameter (SEC1)	U	1 character
Text Security parameters (SEC2))		166 spaces
Encryption (ENCRYP)	0	No Encryption
Text Format (TXTFMT)	STA	3 characters
Extended Subheader Data Length (TXSHDL)	00000	5 digits

E.2 Examples using the Transportable File Structure (TFS)

E.2.1 Use of TFS

The Transportable File Structure (TFS) allows disparate object data, relating to the image, to be included in the BIIF file. This is the mechanism used to contain the PIKS objects for image processing on the image. Additional information about the image and the symbols used in a BIIF file are contained in a TFS. Each Transport in the TFS represents a single entity. For example, each Transport could represent additional patient information for each image in the BIIF file. Each Transport contains one or more Profiles. These Profiles are related together since they are in the same Transport. Since a Profile can contain another Profile, hierarchical information can be included in a TFS. The TFS also allows a user to request data about an image and how to disseminate the BIIF file. Any restrictions or specific use of a TFS should be described clearly in the TFS Profile Proforma. This ensures interoperability between BIIF users.

E.2.2 TFS Examples

The following examples demonstrate different ways in which a TFS could be used in a BIIF file. Each example is presented in text format for clarity and the essential parameters for each TFS command are included.

E.2.2.1 TFS example of PIKS processing for an image

The following TFS example contains one Transport, “Patient X”. This Transport contains two Profiles. The first Profile contains a pointer to the image and the PIKS processing objects for the image. The PIKS objects are to be applied to the image since they all occur within the same Profile. The second Profile contains a pointer to an HTML file.

```
BEGIN TFS “/Profile=BIIF-TFS/ NAME=Patient X Data/ DATE=19970101000000Z/”
    TFS VERSION “1”
    TFS SECURITY “Confidential”
    TFS METADATA “This TFS contains imagery and imagery processing required for Patient X”

    BEGIN TRANSPORT “Patient X”
        TRANSPORT SECURITY “Confidential”
        TRANSPORT METADATA “Data for Patient X”
        BEGIN TRANSPORT BODY

            BEGIN PROFILE “Imagery and processing for Patient X”
                PROFILE SECURITY “Confidential”
                PROFILE METADATA “Imagery, processing for Patient X”
                BEGIN PROFILE BODY
                    PROFILE OBJECT
                     OBJECT_TYPE=2              (Image contained in BIIF segment)
                     OBJECT_NAME=“Image of Patient X”     (name should be meaningful to object)
                     OBJECT_STORAGE=1 (Image stored in TFS or BIIF segment)
                     OBJECT_DATA= (“Name of image in BIIF segment”, 001  (BIIF segment that contains image) )
                PROFILE OBJECT
                     OBJECT_TYPE=1      (Object contents or pointer given below)
                     OBJECT_NAME=“PIKS”      (This is a PIKS object for the above image)
                     OBJECT_STORAGE=1 (PIKS object stored in OBJECT_DATA)
                     OBJECT_DATA=“PIKS_HISTOGRAM”   (PIKS object conforms to Table A.19)
                PROFILE OBJECT
                     OBJECT_TYPE=1      (Object contents or pointer given below)
                     OBJECT_NAME=“PIKS”      (This is a PIKS object for the above image)
                     OBJECT_STORAGE=1 (PIKS object stored in OBJECT_DATA)
                     OBJECT_DATA=“PIKS_LOOKUP_TABLE”   (PIKS object conforms to Table A.19)

                END PROFILE

            BEGIN PROFILE “External data”
                PROFILE SECURITY “Public”
                PROFILE METADATA “HTML file”
                BEGIN PROFILE BODY
                    PROFILE OBJECT “HTML”
                        OBJECT_TYPE=1      (Object contents or pointer given below)
                        OBJECT_NAME=“HTML for Medical Center”     (name should be meaningful to object)
                        OBJECT_STORAGE=2 (Data located at URL given in OBJECT_DATA)
                        OBJECT_DATA= “URL to HTML file”
            END PROFILE

    END TRANSPORT


END TFS
```

E.2.2.2 TFS example of requesting imagery and patient history

The following TFS example contains a subscription for information on Patient X.

```
BEGIN TFS “/Profile=BIIF-TFS/ NAME=Patient X Medical History Request/ DATE=19970101000000Z/”
    TFS VERSION “1”
    TFS SECURITY “Confidential”
    TFS METADATA “This TFS is a request for the medical history for Patient X. Please send the information to the address
```

```
                given in the TFS Subscription command"
    TFS SUBSCRIPTION
        USERNAME="XXXXX"  (login name)
        PASSWORD="YYYYY"  (password if needed)
        HOSTNAME="ZZZZZ"  (host name or IP address if needed)
        URL="ftp://ftp.hospital.com"     (use ftp to transfer file)
        DIRECTORY="/home/patients"       (directory to deposit file)
        FILENAME="patientx.inf"          (file name to use)
        IMAGE=1                          (Patient X image to be included in TFS)
        MEDICAL_RECORD=1                 (Medical record to be included in TFS)
END TFS
```

E.2.2.3 TFS example of using configuration and a five band image

The following TFS example demonstrates how the TFS Configuration command is used.

```
BEGIN TFS "/Profile=BIIF-TFS/ NAME=Remote Sensing Data/ DATE=19970101000000Z/"
    TFS VERSION "1"
    TFS SECURITY "Public Domain"
    TFS METADATA "This TFS contains Remote Sensing data for area X"
    TFS CONFIGURATION=0      (configuration of objects provided in TFS)
    TFS CONFIG DATA
        IMAGE=PIKS      (all objects named IMAGE are PIKS image objects)
        MAPS=TIFF               (all objects named MAPS are in TIFF format)
        INFORMATION=SMGL  (all objects named INFORMATION are in SMGL format)

    BEGIN TRANSPORT "Remote Sensing Data"
        TRANSPORT SECURITY "Public Domain"
        TRANSPORT METADATA "Complete coverage for area X using remote Sensing"
        BEGIN TRANSPORT BODY

            BEGIN PROFILE "Image for area X"
                PROFILE SECURITY "Public Domain"
                PROFILE METADATA "Five band image for area X"
                BEGIN PROFILE BODY
                    PROFILE OBJECT
                        OBJECT_TYPE=1        (Object contents or pointer given below)
                        OBJECT_NAME="IMAGE"            (This is a PIKS image)
                        OBJECT_STORAGE=1         (PIKS image stored in OBJECT_DATA)
                        OBJECT_DATA="PIKS_IMAGE"   (PIKS object conforms to Table A.19)

                            NOTE - If image is contained in a BIIF component, then CPL does not contain the
                            additional byte numbers, and there should be a object pointing to the BIIF
                            component that contains the image.
            END PROFILE

            BEGIN PROFILE "MAP for area X"
                PROFILE SECURITY "Public Domain"
                PROFILE METADATA "Map of area X"
                BEGIN PROFILE BODY
                    PROFILE OBJECT
                        OBJECT_TYPE=1        (Object contents or pointer given below)
                        OBJECT_NAME="MAP"        (This is a PIKS image)
                        OBJECT_STORAGE=1         (PIKS image stored in OBJECT_DATA)
                        OBJECT_DATA=binary data   (MAP of area X in TIFF format)
            END PROFILE

            BEGIN PROFILE "Agriculture for area X"
                PROFILE SECURITY "Public Domain"
                PROFILE METADATA "Agriculture description of area X"
                BEGIN PROFILE BODY
                    PROFILE OBJECT
                        OBJECT_TYPE=1        (Object contents or pointer given below)
```

```
                        OBJECT_NAME="INFORMATION"   (Information is stored in SMGL format)
                        OBJECT_STORAGE=1            (Data stored in OBJECT_DATA)
                        OBJECT_DATA=binary data     (SMGL data)
            END PROFILE
    END TRANSPORT
END TFS
```

E.3 Open Skies Digital Data Exchange Profile example (Informative)

Following are the ANNEXES from *Standardization of Logical Formats for the Exchange of Digital Data among States Parties* to the Open Skies Treaty (27 Nation Members). The annexes from this draft Decision are placed in this document as an example of a planned use of BIIF. It is hoped that these Annexes will be helpful in understanding and using BIIF. The examples in this profile demonstrate how BIIF can be used for Video Frames, Infrared, and Synthetic Aperture Radar (SAR) Imagery.

E.3.1 Open Skies Digital Data Exchange File Header

Table E.10 -- Open Skies Digital Data Exchange File Header

FIELD	VALUE	SIZE	DESCRIPTION	TYPE
FHDR	*OSDE*	4	Profile Name	R
FVER	*01.00*	5	Version	R
CLEVEL	*00*	2	Profile Complexity Level	R
STYPE	*BF01*	4	Standard Type	R
OSTAID	*OPEN SKIES*	10	Originating Source	R
FDT	CCYYMMDDhhmm00	14	Date & Time in Co-ordinated Universal Time per ISO 8601:1988	R
FTITLE	*OPEN SKIES DIGITAL DATA EXCHANGE TAPE ANNOTATION* or *OPEN SKIES DIGITAL DATA EXCHANGE TAPE DIRECTORY* or *OPEN SKIES DIGITAL DATA EXCHANGE IMAGE DATA*(followed by spaces to fill 80 characters)	80	Title	R
FSEC	*FOR OPEN SKIES PURPOSES ONLY* (139 spaces follow)	167	Security Profile Specific Parameters	R
FSCOP	*00000*	5	File Copy Number	R
FSCPYS	*00000*	5	File Total Number of Copies	R
ENCRYP	*0*	1	Encryption (0 = Not Encrypted)	R
OID	State Party Name followed by spaces to fill 42 characters	45	Originator's Identification	R
FL	000000000388 - 999999999999	12	Length of File, including all headers, subheaders and data, in bytes	R
HL	000388 - 999999	6	*OSDE* Header Length, in bytes	R
NUMI	*000* or *001*	3	Number of Images in the File. *000* for Tape Annotation or Tape Directory File, *001* for Image File	R
LISH001	000439 - 999999	6	Length of Image Subheader	C
LI001	0000000001- 9999999999	10	Length of Image Data in Bytes	C
NUMS	*000*	3	Number of Symbols	R
NUMX	*000*	3	Reserved for future data types	R
NUMT	000 - 999	3	Number of Text Elements	R
LTSH001	0282 - 9999	4	Length of 1st Text Subheader	C
LT001	00001 - 99999	5	Length of 1st Text Data	C
...				C
LTSHn	0282 - 9999	4	Length of Nth Text Subheader	C
LTn	00001 - 99999	5	Length of Nth Text Data	C
NUMDES	000 - 999	3	Number of Data Extension Segments	R
LDSH001	0200 - 9999	4	Length of 1st Data Extension Segment Subheader	C

Table E.10 -- Open Skies Digital Data Exchange File Header *(concluded)*

FIELD	VALUE	SIZE	DESCRIPTION	TYPE
LD001	000000001 - 999999999	9	Length of 1st Data Extension Segment Data Field	C
...				C
LDSHn	0200 - 9999	4	Length of Nth Data Extension Segment Subheader	C
LDn	000000001 - 999999999	9	Length of Nth Data Extension Segment Data Field	C
NUMRES	*000*	3	Number of Reserved Extension Segments	R
UDHDL	00000 - 99999	5	User Defined Header Data Length	R
UDHOFL	000 - 999	3	User Defined Header Overflow	C
UDHD	User Defined	*UDHDL*-3	User Defined Header Data	C
XHDL	*00000*	5	Extended Header Data Length	R

E.3.2 Open Skies Image Data Subheader

Table E.11 -- Image Data Subheader

FIELD	VALUE	SIZE	DESCRIPTION	TYPE
IM	*IM*	2	Image Data Subheader Identification	R
IID	Frame Number or Line Counter	10	Image Identification	R
IDATIM	CCYYMMDDhhmm00	14	Date & Time of image in Co-ordinated Universal Time per ISO 8601:1988	R
IINFO	User Defined Image Information (followed by spaces to fill 97 characters)	97	Image Information	R
ISCSEC	*FOR OPEN SKIES PURPOSES ONLY* (139 spaces follow)	167	Image Security Classification	R
ENCRYP	*0*	1	Encryption (0 = Not Encrypted)	R
ISORCE	(Sensor Configuration Number as provided in Notification Formats (CC-RRRR-SSSS) followed by 30 spaces)	42	Image Source	R
NROWS	00000001-99999999	8	Number of Rows in Image	R
NCOLS	00000001-99999999	8	Number of Columns in Image	R
PVTYPE	*INT* or *SI*	3	Pixel Value Type: *INT* is unsigned integer, *SI* is signed integer. (Bits per pixel are shown in field *ABPP*)	R
IREP	*MONO* or *RGB* or *RGB/LUT* (followed by spaces to fill 8 characters)	8	Image Representation	R
ICAT	*VIS* or *IR* or *SAR* or *SARIQ* (followed by spaces to fill 8 characters)	8	Image Category	R
ABPP	01-96	2	Actual Bits-per-Pixel per Band	R
PJUST	*R* or *L*	1	Pixel Justification (*R* = Right Justified *L* = Left Justified)	R
ICORDS	space or *G or N or S or U*	1	Image Co-ordinate System (space = None, G = Geographic, N = Universal Transverse Mercator (UTM) Northern Hemisphere, S = UTM Southern Hemisphere, U = UTM Military Grid Reference System)	R
IGEOLO	user defined	60	Image Location	C
NICOM	*0*	1	Number of Image Comments	R

Table E.11 -- Image data subheader *(continued)*

FIELD	VALUE	SIZE	DESCRIPTION	TYPE
IC	*NC*	2	Image Compression (*NC* = No Compression)	R
NBANDS	1-3	1	Number of Bands comprising the image	R
IREPBAND1	Spaces for IREP=MONO or RGB/LUT, R for IREP=RGB	2	Spaces or R; user defined when *NBANDS* > 3	R
ISUBCAT1	Center Wavelength	6	Center Wavelength of 1st Band (micrometres for visible and infrared imagery; centimeters for SAR)	R
IFC1	*N* or user defined	1	User defined Band 1 Image Filter Condition. Default is *N* (None)	R
IMFLT1	User defined or spaces	3	User Defined Band 1 Standard Image Filter Code. Default is Spaces	R
NLUTS1	0-4	1	Number of Band 1 Look-up Tables (LUTs)	R
NELUT1	00001-65536	5	Present only if *NLUTS1* > 0; number of entries in each Band 1 LUT, when present	C
LUTD11	Look Up Table (LUT) values for 1st LUT of Band 1	*NELUT1*	Present only if *NLUTS1* > 0; LUT entries for 1st LUT of Band 1; This field supports only integer band data (*PVTYPE = INT*)	C
....				C
LUTD1m	LUT values for nth LUT of Band 1	*NELUT1*		C
....				C
IREPBANDn	When IREP=MONO or RGB/LUT IREPBAND2 and IREPBAND3 = Spaces. When IREP=RGB IREPBAND2=G, IREPBAND3=B.	2	n^{th} band Representation This field shall contain a valid indicator of the interpretation of the n^{th} band correlated with the value of IREP.	C
ISUBCATn	Center Wavelength	6	Center Wavelength of nth Band (micrometres for visible and infrared imagery; centimeters for SAR)	C
IFCn	*N* or user defined	1	User defined Band n Image Filter Condition. Default is *N* (None)	C
IMFLTn	User defined or spaces	3	User defined Band n Standard Image Filter Code. Default is Spaces	C
NLUTSn	0-4	1	Number of Band n LUTs	C
NELUTn	00001-65536	5	Present only if *NLUTSn* > 0; number of entries in each Band n LUT, when present	C

Table E.11 – Image data subheader *(concluded)*

FIELD	VALUE	SIZE	DESCRIPTION	TYPE
LUTDn1	Look Up Table (LUT) values for 1st LUT of Band n	*NELUTn*	Present only if *NLUTSn* > 0; LUT entries for 1st LUT of Band nn; This field supports only integer band data (*PVTYPE = INT*)	C
....				C
LUTDnm	LUT values for mth LUT of Band n	*NELUTn*		C
ISYNC	0-9	1	0 to indicate no synchronisation codes in image data; other values user defined	R
IMODE	*P* or *S*	1	*P* = pixel interleaved (RGB or IQ pixels); *S* = Band Sequential (B image follows G image follows R image, or I “image” follows Q “image”. Use *P* for single band images	R
NBPR	0001-9999	4	Number of blocks per row	R
NBPC	0001-9999	4	Number of blocks per column	R
NPPBH	0001-9999	4	Number of pixels per block horizontal. Any combination of *NBPR* and *NPPBH* such that *NBPR***NPPBH* >= *NCOLS* is acceptable	R
NPPBV	0001-9999	4	Number of pixels per block vertical. Any combination of *NBPC* and *NPPBV* such that *NBPC***NPPBV* >= *NROWS* is acceptable	R
NBPP	01-96	2	Number of storage bits per pixel per band *NBPP* >= *ABPP*	R
IDLVL	*001*	3	Display Level	R
IALVL	*000*	3	Attachment Level	R
ILOC	*0000000000*	10	Image Location	R
IMAG	*1.00*	4	Image Magnification	R
UDIDL	00000 or 00003- 99999	5	User Defined Image Data Length (Length in bytes of the entire field *UDID* plus three bytes)	R
UDOFL	000 - 999	3	User Defined Overflow	C
UDID	(See Annex C for SAR Information Parameters)	*UDIDL*-3	SAR Information Parameters and a Description of Ancillary Data for SAR or Other Sensors	C (*UDID L>0*)
IXSHDL	00000 or 00003-99999	5	Extended Subheader Data Length (Length of *IXSHD* plus 3)	R
IXSOFL	000 - 999	3	Extended Subheader Overflow	C
IXSHD	User Defined	*IXSHDL*-3	Extended Subheader Data	C

E.3.3 Open Skies SAR Information

Table E.12 -- Tagged Record Extension for SAR Information Parameters

FIELD	VALUE	SIZE	DESCRIPTION	TYPE
TRETAG	cc*SAR*n	6	Unique extension type identifier for SAR information parameters, where cc is two letter country code and n is user assigned format number	R
TREL	00076-99985	5	Total Length of SAR information parameter description fields	R
TREDATA	SAR Information Parameters Table C2	*	User Defined for SAR data	R

* User defined field size

Table E.13 -- SAR Information Parameters

FIELD	VALUE	SIZE	DESCRIPTION	TYPE
SARTYP	*LINEAR FM CHIRP* (5 spaces follow)	20	SAR Type	R
SARRT	*R* or *T*	1	*SARSLANTMN* Units, *R*= Range to first sample, *T*= Time to first sample	R
SARSLANTMN	000.0-999.9	5	Range/Time to First Sample	R
SARFW	*F* or *W*	1	*SAROPFREQ* Units, *F*= Frequency, *W*= Wavelength	R
SAROPFREQ	00000.00-99999.99	8	Emitted Pulse Carrier Frequency or Wavelength	R
SARBANDTX	000.00-999.99	6	Emitted Pulse Bandwidth	R
SARDUR	00.0000-99.9999	7	Emitted Pulse Duration	R
SARNP	*N* or *P*	1	*SARPULSES* Units *N*= number of pulses emitted in pulses per metre of flight path, *P*= Pulse Repetition Frequency in Hz	R
SARPULSES	0000.000-9999.999	8	Emitted Number of Pulses	R
SARVEL	000.0000-999.9999	8	Along-track Platform Velocity	R
SARAAB	0.0000-9.9999	6	Azimuth Antenna Beamwidth	R
SARRANNUM	00000-99999	5	Number of Range Samples	R
...	(SAR-specific Data) User Defined	*	SAR-specific Data User Defined	C (*REL>76*)

* User defined field size

TRETAG
This field contains a unique extension type identifier for SAR information parameters, where cc is two letter country code and n is user assigned format number to allow for additional formats in the future.

TREL
This field contains the length in bytes of the data contained in *TREDATA*.

TREDATA
This field contains user defined data.

SARTYP
This field indicates the type of SAR that collected the data in this file. At the present time, the only valid value is *LINEAR FM CHIRP* which indicates a straight-line flying SAR using a linear FM chirp emitted pulse.

SARRT
This field indicates whether the range to first sample or the time to first sample is stored in the *SARSLANTMN* field. If this field contains *R*, then *SARSLANTMN* contains the range to first sample. Otherwise, this field contains *T* and *SARSLANTMN* contains the time to first sample.

SARSLANTMN
If (*SARRT* = *R*), then this field contains the range to first sample in metres. Otherwise, this field contains the time to first sample in microseconds.

SARFW
This field indicates whether SAR operating frequency or SAR operating wavelength is contained in the *SAROPFREQ* field. If this field contains *F*, the *SAROPFREQ* contains the SAR operating frequency. Otherwise, this field contains *W* and *SAROPFREQ* contains the SAR operating wavelength.

SAROPFREQ
If (*SARFW* = *F*), then this field contains the emitted pulse carrier frequency in MHz to the nearest one-tenth Megahertz. Otherwise, this field contains the emitted pulse wavelength in centimetres.

SARBANDTX
This field contains the emitted pulse bandwidth in MHz.

SARDUR
This field contains the emitted pulse duration in microseconds.

SARNP
This field indicates whether number of pulses or pulse repetition frequency is stored in the *SARPULSES* field. If this field contains *N*, then *SARPULSES* contains the number of pulses emitted in pulses per metre of flight path. Otherwise, this field contains *P*, and *SARPULSES* contains the pulse repetition frequency.

SARPULSES
If (*SARNP* = *P*), then this field contains the number of pulses emitted in pulses per metre of flight path. Otherwise, this field contains the emitted pulse repetition frequency in Hz.

SARVEL
This field contains the along-track platform velocity in metres per second.

SARAAB
This field contains the azimuthal antenna beamwidth in radians.

SARRANNUM
This field contains the number of range samples generated by the radar per metre of slant range.

E.3.4 Open Skies Text Data Subheader

Table E.14 -- Text Data Subheader

FIELD	VALUE	SIZE	DESCRIPTION	TYPE
TE	*TE*	2	Text Subheader Identifier	R
TEXTID	*ANNOTATION* or *OSDDEF DIR* or *TAPEHEADER*	10	Text Identification	R
TXTDT	CCYYMMDDhhmm00	14	Date & Time of image in Co-ordinated Universal Time per ISO 8601:1988	R
TXTITL	*OPEN SKIES TAPE ANNOTATION* or *OPEN SKIES TAPE DIRECTORY* or *OPEN SKIES IMAGE ANNOTATION* (followed by spaces to fill 80 characters)	80	Text Title	R
TSSEC	*FOR OPEN SKIES PURPOSES ONLY* (139 spaces follow)	167	Text Security Classification	R
ENCRYP	*0*	1	Encryption (0 = not encrypted)	R
TXTFMT	user defined	3	Identifier for Text Format for Annotation Entries, default is *STA* for ASCII	R
TXSHDL	00000-09717	5	Extended Text Subheader Data Length	R
TXSHD	Used Defined	*TXSHDL*	Extended Subheader Data	C

TE
This field shall have the value *TE* to identify the subheader as a text subheader.

TEXTID
This field contains the text identification. The value shall be *ANNOTATION* for Open Skies Digital Data Exchange Format image files, *OSDDEF DIR* for Open Skies Digital Data Exchange Format Tape Directory files, or *TAPEHEADER* for Open Skies Digital Data Exchange Tape Annotation files.

TXTDT
This field contains the date and time in accordance with the provisions of ISO 8601:1988 for expressing combined date and time of day representations of file creation in the format CCYYMMDDhhmm00, where CCYY is the four digit year , MM is the month (01-12), DD is the day of the month (01-31), hh is the hour (00-23), mm is the minute (00-59), and 00 in place of seconds to express the time of day in Co-ordinated Universal Time (UTC).

TXTITL
This field contains the title of the text item.

TSSEC
This field contains the profile specific information for the text product security and is defined in the profile.

ENCRYP
This field contains the encryption code as required by the BIIF format. The value shall be *0* (not encrypted).

TXTFMT
This field contains a three character code that can be used to indicate the user-defined format or template to be used to display the text. The template and its three character code shall be provided as part of the sensor information made available at the time of certification. In the event that additions or changes not requiring recertification of the sensor are made to a user-defined text format after certification has been completed, notification of such changes shall be provided to the Open Skies Consultative Commission. The default value for *TXTFMT* is *STA*, indicating that the text values that follow are in Standard ASCII format.

TXSHDL
This field contains the length in bytes of the *TXSHD*. A value of zero shall mean that no profile defined tagged record extensions are included in the text subheader. If a profile defined tagged record extension is too long to fit in the *TXSHD* field, it shall be put in a data extension segment (DES).

TXSHD
This field contains the user defined tagged record extensions. The value of the first three bytes of this field shall be *000* if this field does not overflow into a DES, or it shall contain the sequence number of the DES into which it does overflow.

E.3.5 Open Skies Annotation Text Format [1]

Table E.15 -- Treaty on Open Skies Annotation Text Format

FIELD	VALUE	SIZE	DESCRIPTION	TYPE
OSFLT	*OS*XYYY	6	Observation Flight Reference Number	R
OSDAT	CCYYMMDD	8	Observation Flight Date	R
OSSNSR	XXXXYY	6	Sensor Description	R
SENSINSTAL	AAA-B-C-DD	10	Sensor Installation	R
OSFCLL	000-999	3	Focal Length in millimetres	R
OSDTG	CCYYMMDDhhmm	12	Image Date & Time	R
OSHAGL	XXXXXY	6	Height Above Ground Level	R
OSLOC	dd.dd(*N* or *S)* ddd.dd(*E* or *W)* or dd mm(*N* or *S)* ddd mm(*E* or *W)*	14	Aircraft Location	R
OSHDG	000-359	3	Aircraft True Heading	R
OSSCAN	000-359	3	Scan Angle; *000* for *OSSNSR* = *SAR*...	R
OSLDA	00-90	2	Look Down Angle to the nearest point of the swath width for *OSSNSR* = *SAR*...; 00 otherwise	R
OSNEAR	00-99	2	The nearest point of the swath width in kilometres for *OSSNSR* = *SAR...; 00* otherwise	R
OSSWTH	000-999	3	Swath Width for *OSSNSR* = *SAR...; 000* otherwise	R
OSPOL	*HH* or *HV* or *VH* or *VV* or spaces	2	Polarisation for *OSSNSR* = *SAR...;* spaces otherwise	R
OSSPD	XXXYY	5	Ground Speed for *OSSNSR* = *SAR...; 000YY* or Ground Speed otherwise	R
OSDRFT	XXY	3	Drift for *OSSNSR* = *SAR...; 00Y* or Drift otherwise	R
OSPTCH	XXY	3	Pitch Angle for *OSSNSR* = SAR...; 00Y or Pitch Angle otherwise	R
OSROLL	XXY	3	Roll Angle for *OSSNSR* = *SAR...; 00Y* or Roll Angle otherwise	R
OSADDL	00000-99898	5	Length of Additional Items in bytes	R
OSADDAN	(User Defined)	User defined between 0-99898	Additional Annotation Data (User Defined)	C (*OSADDL*>0)

OSFLT
This field contains the reference number of the observation flight during which the image data was collected. The format of the field is described in Appendix 1 to Treaty Annex B.

OSDAT
This field contains the date of the observation flight in Co-ordinated Universal Time in the format CCYYMMDD, where CCYY is the four digit year, MM is the month (01-12) and DD is the day of the month (01-31).

OSSNSR

[1] Note: Open Skies “Annotations” refer to text metadata. This is not the same as BIIF Symbol “Annotations”

This field contains sensor description information in the format specified in Appendix 1 to Treaty Annex B.

SENSINSTAL
This field contains the 10 character sensor installation number (AAA-B-C-DD) identical to that used in the Open Skies Notification Formats 4, 5, and 6, where AAA is *INT* or *POD* to indicate internal or podded installation, B is a number to indicate relative position of from nose to tail for internal sensors or pod mounting location for podded sensors (*L* = mounted under left wing, *R* = mounted under right wing, *C* = mounted on aircraft centre-line), C is type of installation (*V* = vertical, *L* = left, *R* = right, F = fan of two or more sensors, DD is depression angle in degrees for vertical and oblique sensors, or for fan installations, total number of sensors followed by individual sensor number in sequence from left to right relative to direction of flight.

OSFCLL
This field contains the focal length of the sensor in millimetres. If focal length is not applicable, then the value of this field is *000*.

OSDTG
This field contains the date and time that the data was collected to the nearest minute of Co-ordinated Universal Time. The format of this field is CCYYMMDDhhmm, where CCYY is the four digit year, MM is the month (01-12) and DD is the day of the month (01-31), hh is the hour (00-23), mm is the minute (00-59).

OSHAGL
This field contains the average height above ground level of the observation aircraft. The format of this field is XXXXXY, where XXXXX is a 5-digit number and Y is a 1-letter code representing the units of measurement; valid codes are *F* indicating units of feet, or *M* indicating units of metres.

OSLOC
This field contains the latitude and longitude of the position of the observation aircraft in units of degrees to the nearest one-hundredth of a degree in the format "dd.dd(*N* or *S)* ddd.dd(*E* or *W)*" or in units of degrees and minutes to the nearest minute in the format "dd mm(*N* or *S)* ddd mm(*E* or *W)*".

OSHDG
This field contains the true heading of the observation aircraft in units of degrees to the nearest degree.

OSSCAN
If the sensor category is SAR (i.e., *OSSNSR* = *SAR*XYY), then this field contains the value *000*. Otherwise, this field contains the scan angle of the sensor in degrees.

OSLDA
If the sensor category is not SAR (i.e., *OSSNSR* ≠ *SAR*XYY), then this field contains the value *00*. Otherwise, this field contains the look down angle to the nearest point of the swath width in units of degrees measured from the vertical.

OSNEAR
If the sensor category is not SAR (i.e., *OSSNSR* ≠ *SAR*XYY), then this field contains the value *00*. Otherwise, this field contains the ground distance to the nearest point of the swath width in units of kilometres.

OSSWTH
If the sensor category is not SAR (i.e., *OSSNSR* ≠ *SAR*XYY), then this field contains the value *000*. Otherwise, this field contains the swath width measured in units of kilometres.

OSPOL
If the sensor category is not SAR (i.e., *OSSNSR* ≠ *SAR*XYY), then this field contains spaces. Otherwise, this field contains the polarisations of the radar. Valid entries are *HH*, *HV*, *VH*, and *VV*.

OSSPD
If the sensor category is not SAR (i.e., *OSSNSR* ≠ *SAR*XYY), then this field shall contain the value *000YY*, or optionally, the ground speed of the observation aircraft. For SAR, this field shall contain the ground speed of the observation aircraft. The format of the field is XXXYY, where XXX is a 3-digit number and YY is a 2-letter code representing the units of measurement; valid codes are *NM* indicating nautical miles per hour, or *KM* indicating kilometres per hour.

OSDRFT
If the sensor category is not SAR (i.e., *OSSNSR* ≠ *SAR*XYY), then this field shall contain the value *00Y,* or optionally, the drift angle of the observation aircraft. For SAR, this field shall contain the drift angle of the observation aircraft in units of degrees. The format of this field is XXY where XX is a 2-digit number and Y is a 1-letter code representing the direction of drift; valid codes are *L* indicating drift is to the left, or *R* indicating drift is to the right, relative to the flight path of the observation aircraft.

OSPTCH
If the sensor category is not SAR (i.e., *OSSNSR* ≠ *SAR*XYY), then this field shall contain the value *00Y*, or optionally, the pitch angle of the observation aircraft. For SAR, this field shall contain the pitch angle of the observation aircraft in units of degrees. The format of this field is XXY where XX is a 2-digit number and Y is a 1-letter code representing the direction of pitch; valid codes are *U* indicating pitch is up, or *D* indicating pitch is down, relative to the horizontal.

OSROLL
If the sensor category is not SAR (i.e., *OSSNSR* ≠ *SAR*XYY), then this field shall contain the value *00Y*, or optionally, the roll angle of the observation aircraft. For SAR, this field shall contain the roll angle of the observation aircraft in units of degrees. The format of this field is XXY where XX is a 2-digit number and Y is a 1-letter code representing the direction of pitch; valid codes are *L* indicating roll is to the left, or *R* indicating roll is to the right.

OSADDL
This field contains the length in bytes of the data contained in *OSADDAN*.

OSADDAN
This field contains additional annotation information that is allowed by Decision 9. The length of this field shall not cause any BIIF field length limits to be exceeded, but is otherwise fully definable for each annotation data system. If the additional data is too long to fit in the *UDID* field, it shall be put in a data extension segment.

E.3.6 Data Extension Segment Subheader

Table E.16 -- Data Extension Segment Subheader

FIELD	VALUE	SIZE	DESCRIPTION	TYPE
DE	*DE*	2	Data Extension Segment Subheader Identifier	R
DESID	*TRE_OVERFLOW* or *TRANSPORTABLE_FILE_STRUCT* (followed by spaces)	25	Unique DES Type Identifier	R
DESVER	01-99	2	Version of data field definition	R
DESCLAS	*FOR OPEN SKIES PURPOSES ONLY* (followed by 139 spaces)	167	DES Security Classification	R
DESOFLW	*UDHD or UDID or IXSHD or TXSHD* (followed by spaces), otherwise omitted	6	Identifies extension field which overflows into DES	C
DESITEM	*000-999*	3	Number of the data item overflowing into DES.	C
DESSHL	*0000*	4	Length of DESSHF; Required to be *0000* when *DESID = TRE_OVERFLOW*	R
DESSHF	(User Defined)	User Defined	User Defined Subheader Fields. Not Present if *DESSHL = 0000*	C
DESDATA	(User Defined)	User Defined	User Defined Data Field	R

E.3.7 Open Skies Digital Data Exchange Format (OSDDEF) DCRsi TAPE ANNOTATION and TAPE DIRECTORY File Examples

A. Example OSDDEF Header for OSDDEF TAPE ANNOTATION file on a DCRsi tape containing 245 OSDDEF image files from a video camera.

Table E.17 -- Tape annotation file example

Field	Size	Value
FHDR	4	OSDE
FVER	5	01.00
CLEVEL	2	00
STYPE	4	BF01
OSTAID	10	OPEN SKIES
FDT	14	19961002103000
FTITLE	80	OPEN SKIES DIGITAL DATA EXCHANGE TAPE ANNOTATION
FSEC	167	FOR OPEN SKIES PURPOSES ONLY
FSCOP	5	00000
FSCPYS	5	00000
ENCRYP	1	0
OID	45	USA
FL	12	000000000722
HL	6	000397
NUMI	3	000
NUMS	3	000
NUMX	3	000
NUMT	3	001
LTSH001	4	0282
LT001	5	00043
NUMDES	3	000
NUMRES	3	000
UDHDL	5	00000
XHDL	5	00000
	397	

B. Example OSDDEF Text Subheader for OSDDEF TAPE ANNOTATION file on a DCRsi tape containing 245 OSDDEF image files from a video sensor.

Table E.18 -- Tape annotation file example

Field	Size	Value
TE	2	TE
TEXTID	10	TAPEHEADER
TXTDT	14	19961002103000
TXTITL	80	OPEN SKIES TAPE ANNOTATION
TSSEC	167	FOR OPEN SKIES PURPOSES ONLY
ENCRYP	1	0
TXTFMT	3	STA
TXSHDL	5	00000
	282	

C. Example OSDDEF Text Data for OSDDEF TAPE ANNOTATION file on a DCRsi tape containing 245 OSDDEF image files from a video sensor

Table E.19 -- Tape annotation file

```
1        2        3        4        5
123456789012345678901234567890123456789012345678901234567890123
OS5423
19960507
TVTD
INT-2-V-90
120
```

D. Example OSDDEF Header for OSDDEF TAPE DIRECTORY file on a DCRsi tape containing 245 OSDDEF directory entries from a video sensor

Table E.20 -- Tape Directory File Example

Field	Size	Value
FHDR	4	OSDE
FVER	5	01.00
CLEVEL	2	00
STYPE	4	BF01
OSTAID	10	OPEN SKIES
FDT	14	19961002103000
FTITLE	80	OPEN SKIES DIGITAL DATA EXCHANGE TAPE DIRECTORY
FSEC	167	FOR OPEN SKIES PURPOSES ONLY
ENCRYP	1	0
OID	45	USA
FL	12	000000014654
HL	6	000397
NUMI	3	000
NUMS	3	000
NUMX	3	000
NUMT	3	001
LTSH001	4	0282
LT001	5	13975
NUMDES	3	000
NUMRES	3	000
UDHDL	5	00000
XHDL	5	00000
	387	

E. Example OSDDEF Text Subheader for OSDDEF TAPE DIRECTORY file on a DCRsi tape containing 245 OSDDEF directory entries from a video sensor

Table E.21 -- Tape Directory File

Field	Size	Value
TE	2	TE
TEXTID	10	OSDDEF DIR
TXTDT	14	19961002103000
TXTITL	80	OPEN SKIES TAPE DIRECTORY
TSSEC	167	FOR OPEN SKIES PURPOSES ONLY
ENCRYP	1	0
TXTFMT	3	STA
TXSHDL	5	00000
	282	

F. Example OSDDEF Text Data for OSDDEF TAPE DIRECTORY file on a DCRsi tape containing 245 OSDDEF image files

The Text Data consists of a single line of text containing the number of OSDDEF image files on the DCRsi tape (8 characters + Carriage Return + Line Feed), followed by one line of text for no fewer than each OSDDEF image file on the tape for which there is a change in annotation from the preceding OSDDEF image file in the sensor or time of collection.. Each OSDDEF tape directory file entry consists of:

(A) a 12 character date and time of image in Co-ordinated Universal Time. (columns 1-12)
(B) a 6 character Sensor Description, consisting of a group of up to 4 characters specifying the category of the sensor followed by a 2 character group representing the type of recording medium. (columns 13-18)
(C) a 12 character Sensor Configuration Number (CC-RRRR-SSSS) identical to that used in Formats 4, 5, 6, 8 and 25 (columns 19-30)
(CC = two letter country code; RRRR = "TVLI", "TVFI", "IRLS" or "SAR "; SSSS = nationally assigned unique string of 4 digits)
(D) a 3 numeric character Focal Length, if applicable. (columns 31-33)
(E) a 14 character Aircraft Location in decimal degrees (dd.dd(N or S) ddd.dd(E or W)) or degrees-minutes (dd mm(N or S) ddd mm(E or W)). (columns 34-47)
(F) an 8 numeric character Starting DCRsi Scan Number, identifying the DCRsi Scan Number at which the OSDDEF image file begins. (columns 48-55)
(G) Carriage Return + Line Feed. (columns 56-57)

Table E.22 -- Data Field Table Example

```
1       2       3       4       5       6       7
12345678901234567890123456789012345678901234567890123456789012345678901234567
00000245
199605071207TVTD  US-TVLI-807612043.67N 017.45E00200000
199605071208TVTD  US-TVLI-807612043.67N 017.45E00200475
199605071209TVTD  US-TVLI-807612043.67N 017.45E00200957
199605071210TVTD  US-TVLI-807612043.67N 017.45E00201422
                                        .
                                        .
```

E.3.8 Example Entries in Data and Field Tables

A. Examples of recorded information:

1. **TV1**: Image obtained from a 512 x 512 frame format, black & white video imaging system;
2. **TV2:** Image from a line scanning (6000 elements per line) colour video imaging system;
3. **IR:** Image from a line scanning infrared detector (13,000 elements per line).
4. **SARIMG:** Image obtained from a synthetic aperture radar (13,000 elements / slant range).
5. **SARIQ:** Radio hologram (initial phase information) from a synthetic aperture radar with 13,000 elements / slant range.

Each file contains 512 image lines.

B. Example file header (Tables in Clause E.3.1)

Table E.23 -- Header example for various exchange files examples

Field	Size	TV1	TV2	IR	SARIMG	SARIQ	Rem.
FHDR	4	OSDE	OSDE	OSDE	OSDE	OSDE	
FVER	5	01.00	01.00	01.00	01.00	01.00	
CLEVEL	2	00	00	00	00	00	
STYPE	4	BF01	BF01	BF01	BF01	BF01	
OSTAID	10	OPEN SKIES	OPEN SKIES	OPEN SKIES	OPEN SKIES	OPEN SKIES	
FDT	14	19961002103000	19961002103000	19961002103000	19961002103000	19961002103000	
FTITLE	80	OPEN SKIES DIGITAL DATA EXCHANGE IMAGE DATA	OPEN SKIES DIGITAL DATA EXCHANGE IMAGE DATA	OPEN SKIES DIGITAL DATA EXCHANGE IMAGE DATA	OPEN SKIES DIGITAL DATA EXCHANGE IMAGE DATA	OPEN SKIES DIGITAL DATA EXCHANGE IMAGE DATA	
FSEC	167	FOR OPEN SKIES PURPOSES ONLY	FOR OPEN SKIES PURPOSES ONLY	FOR OPEN SKIES PURPOSES ONLY	FOR OPEN SKIES PURPOSES ONLY	FOR OPEN SKIES PURPOSES ONLY	
FSCOP	5	00000	00000	00000	00000	00000	
FSCPYS	5	00000	00000	00000	00000	00000	
ENCRYP	1	0	0	0	0	0	
OID	45	RUSSIA	RUSSIA	RUSSIA	RUSSIA	RUSSIA	
FL	12	000000263366	000009217248	000006657222	000006657222	000013325647	
HL	6	000417	000417	000417	000417	000417	
NUMI	3	001	001	001	001	001	
LISH001	6	000422	000448	000422	000422	000559	
LI001	10	0000262144	0009216000	0006656000	0006656000	0013324288	
NUMS	3	000	000	000	000	000	
NUMX	3	000	000	000	000	000	
NUMT	3	001	001	001	001	001	
LTSH001	4	0282	0282	0282	0282	0282	
LT001	5	00101	00101	00101	00101	00101	

Table E.23 -- Header example for various exchange files examples (*concluded*)

Field	Size	TV1	TV2	IR	SARIMG	SARIQ	Rem.
NUMDES	3	000	000	000	000	000	
LDSH001	4	0000	0000	0000	0000	0000	
NUMRES	3	000	000	000	000	000	
UDHDL	5	00000	00000	00000	00000	00000	
XHDL	5	00000	00000	00000	00000	00000	

417

C. Example image data subheader (Tables in Clause E.3.2)

Table E.24 -- Image data subheader

Field	Size	TV1	TV2	IR	SARIMG	SARIQ	Rem.
IM	2	IM	IM	IM	IM	IM	
IID	10	0000000001	0000000001	0000000001	0000000001	0000000001	Frame No. or Line Counter
IDATIM	14	19961002103000	19961002103000	19961002103000	19961002103000	19961002103000	
IINFO	97	OPEN SKIES IMAGE	OPEN SKIES IMAGE	OPEN SKIES IMAGE	OPEN SKIES IMAGE	OPEN SKIES IMAGE	User Defined image information
ISCSEC	167	FOR OPEN SKIES PURPOSES ONLY	FOR OPEN SKIES PURPOSES ONLY	FOR OPEN SKIES PURPOSES ONLY	FOR OPEN SKIES PURPOSES ONLY	FOR OPEN SKIES PURPOSES ONLY	
ENCRYP	1	0	0	0	0	0	
ISORCE	42	RF-TVFI-0001	RF-TVLI-0001	RF-IRLS-0001	RF-SAR -0001	RF-SAR -0001	Sensor Configuration Number
NROWS	8	00000512	00000512	00000512	00000512	00000512	
NCOLS	8	00000512	00006000	00013000	00013000	00013000	
PVTYPE	3	INT	INT	INT	INT	SI	
IREP	8	MONO	RGB	MONO	MONO	MONO	
ICAT	8	VIS	VIS	IR	SAR	SARIQ	
ABPP	2	08	08	08	08	16	
PJUST	1	R	R	R	R	R	
ICORDS	1	(space)	(space)	(space)	(space)	(space)	
NICOM	1	0	0	0	0	0	
IC	2	NC	NC	NC	NC	NC	
NBANDS	1	1	3	1	1	1	
IREPBAND 1	2	(2 spaces)	R	(2 spaces)	(2 spaces)	(2 spaces)	
ISUBCAT1	6	00.530	00.630	10.000	04.000	04.000	
IFC1	1	N	N	N	N	N	
IMFLT1	3	(3 spaces)	(3 spaces)	(3 spaces)	(3 spaces)	(3 spaces)	
NLUTS1	1	0	0	0	0	0	
IREPBAND 2	2		G				
ISUBCAT2	6		00.530				
IFC2	1		N				
IMFLT2	3		(3 spaces)				
NLUTS2	1		0				
IREPBAND 3	2		B				
ISUBCAT3	6		00.450				
IFC3	1		N				
IMFLT3	3		(3 spaces)				
NLUTS3	1		0				
ISYNC	1	0	0	0	0	0	
IMODE	1	P	P	P	P	P	
NBPR	4	0001	0001	0002	0002	0002	
NBPC	4	0001	0001	0001	0001	0001	
NPPBH	4	0512	6000	6500	6500	6500	
NPPBV	4	0512	0512	0512	0512	0512	
NBPP	2	08	08	08	08	16	

Table E.24 -- Image data subheader (*concluded*)

Field	Size	TV1	TV2	IR	SARIMG	SARIQ	Rem.
IDLVL	3	001	001	001	001	001	
IALVL	3	000	000	000	000	000	
ILOC	10	0000000000	0000000000	0000000000	0000000000	0000000000	
IMAG	4	1.00	1.00	1.00	1.00	1.00	
UDIDL	5	00000	00000	00000	00000	00137	
UDOFL	3					000	
UDID						Table 4	
IXSHDL	5	00000	00000	00000	00000	00000	

D. Example OSDDEF Text Subheader

Table E.25 -- Text subheader

Field	Size	Contents
TE	2	TE
TEXTID	10	ANNOTATION
TXTDT	14	19961002103000
TXTITL	80	OPEN SKIES IMAGE ANNOTATION
TSSEC	167	FOR OPEN SKIES PURPOSES ONLY
ENCRYP	1	0
TXTFMT	3	STA
TXSHDL	5	00000
	282	

E. Example field record expansions for the SAR initial phase data (Annex C Tables)

Table E.26 -- RETAG data for SAR initial phase data

Field	Size	Value	Comments
TRETAG	6	RUSAR1	
TREL	5	00123	
TREDATA	123	User Defined Registered Extension for SAR Data Table 5	

F. User Defined Registered Extension For SAR Data.

Table E.27 -- User defined registered extension

Field	Size	Value	Comments
SARTYP	20	LINEAR FM CHIRP	
SARRT	1	R	
SARSLANTMN	5	04000	
SARFW	1	F	
SAROPFREQ	8	09000.00	
SARBANDTX	6	040.00	
SARDUR	7	10.0000	
SARNP	1	P	
SARPULSES	8	2000.000	
SARVEL	8	010.0000	
SARAAB	6	0.0523	
SARRANNUM	5	00001	
SARNL	5	00042	Length of Description
SARNDAT	42	Table 6	Synthetic aperture radar micronavigational parameters
	123		

G. Expansion of synthetic aperture radar micronavigation parameters from Table 5 (SARNDAT data)

Table E.28 -- Expansion of SAR micronavigational parameters

Field	Size	Value	Comments
SARNDAT11	6	Vx	X-axis velocity
SARNDAT12	1	4	field size in bits
SARNDAT21	6	Vy	Y-axis velocity
SARNDAT22	1	4	field size in bits
SARNDAT31	6	Vz	Z-axis velocity
SARNDAT32	1	4	field size in bits
SARNDAT41	6	ax	X-axis acceleration
SARNDAT42	1	4	field size in bits
SARNDAT51	6	ay	Y-axis acceleration
SARNDAT52	1	4	field size in bits
SARNDAT61	6	az	Z-axis acceleration
SARNDAT62	1	4	field size in bits
	42		

Configuration of synthetic aperture radar phase data corresponding to Table 6

Table E.29 -- Configuration of SAR

Size	4	4	4	4	4	4	8	8	8	8	Field size in bits
Value	Vx	Vy	Vz	ax	ay	az	I	Q	I	Q	Line phase data

13000*2*8=208000

A second example configuration of synthetic aperture radar phase data.

Table E.30 -- 2nd example of configuration of SAR

FIELD	Size	Value	Description
	128		SAR specific Header Data
	26624		SAR IQ Data

Sum: 26752

H. Example Data Extension Segment (DES). This table is constructed assuming there is overflow from a tagged record extension field into a DES. The field DESDATA is user defined in both size and value and would be dependent on the tagged record extension which overflowed from the UDHD field.

Table E.31 -- Data extension segment for SAR

Field	Size	TV1	TV2	IR	SARIMG	SARIQ	Rem
DE	2	DE	DE	DE	DE	DE	
DESID	25	TRE_OVERFLOW	TRE_OVERFLOW	TRE_OVERFLOW	TRE_OVERFLOW	TRE_OVERFLOW	
DESVER	2	01	01	01	01	01	
DESCLAS	167	FOR OPEN SKIES PURPOSES ONLY	FOR OPEN SKIES PURPOSES ONLY	FOR OPEN SKIES PURPOSES ONLY	FOR OPEN SKIES PURPOSES ONLY	FOR OPEN SKIES PURPOSES ONLY	
DESOFLW	6	UDID	UDID	UDID	UDID	UDID	
DESITEM	3	001	001	001	001	001	
DESSHL	4	0000	0000	0000	0000	0000	
DESDATA	TBD	user defined	user defined	user defined	user defined	user defined	

209+TBD (To Be Defined)